



## Source Code Sector Model

*CANVAS Group*

*Authors: ME Clark, PhD, Meide Zhao, PhD,  
m-clark4@uiuc.edu, mzhao@uic.edu  
September, 1999*

C COPYRIGHT JANUARY, 1999 THE BOARD OF TRUSTEES AT THE UNIVERSITY OF ILLINOIS  
 C ALL RIGHTS RESERVED  
 C AUTHORED BY M.E.CLARK AND M.ZHAO

C

C PROGRAM CZ (CLARK/ZHAO) MODEL

C

C DATES OF REVISIONS AND MODIFICATIONS

C REV 12/9/98 TO ALLOW UTILIZATION OF MAGNETIC RESONANCE FLOW DATA

C REV 1/10/98 TO ALLOW EITHER PRESSURE OR FLOW SOURCES

C REV 7/9/87 TAKE OUT CHAR IMP FOR 36, MODIFIED 6/16/88 W/FJE

C REVISED TO PUT CHAR.IMP. AT END OF 36 FOR ARM/HAND/SYS MODEL 6/19/87

C SET UP FOR AUTO CUTOFF ON PRESS.LT.PSTOP

C MODIFIED 8/90 JUNC SUBSC

C

C

C PROGRAM TO STUDY DISTRIBUTION OF BLOOD WITHIN A SYSTEM OF

C VESSELS. NECESSARY INPUT PARAMETERS ARE REQUIRED TO DESCRIBE

C THE STRUCTURAL LAYOUT AND DEFINE PHYSIOLOGICAL PARAMETERS.

C RESULTS ARE OUTPUTTED TO EXTERNAL FILES AND PLOTS ARE PRODUCED.

C PRESSURES OUTPUT IN MMHG, FLOWS IN CC/SEC

C PROGRAM PARAMETERS NPS=NPSAVE

IMPLICIT REAL\*8 (A-H,O-Z)

INTEGER TUBES, POINTS, TERMS, FORCE, OTHER, FORCQ, FORCP, FORCT

INTEGER TQLINK, RLINK, TLINK, PLINK, QLINK, FLAG, TIMP

PARAMETER (TUBES=121, POINTS=14, TERMS=17, OTHER=250, NPS=14)

PARAMETER (FORCQ=1, FORCP=1, FORCT=1)

REAL\*4 BB (POINTS)

DIMENSION PFG (FORCP, 50), QFG (FORCQ, 64), TFG (FORCT, 64)

DIMENSION PMULT (FORCP), TLAGP (FORCP), QMULT (FORCQ), TLAGQ (FORCQ)

DIMENSION TMULT (FORCT), TLAGT (FORCT), MODEL62 (TUBES), TQLINK (TUBES)

DIMENSION P (TUBES, POINTS), Q (TUBES, POINTS), PH (TUBES, POINTS)

DIMENSION A (TUBES, POINTS), AO (TUBES, POINTS), CAP (TUBES, POINTS)

DIMENSION LQ (TUBES), LP (TUBES), RTOT (TERMS), CVTER (TERMS), ZC (TUBES)

DIMENSION R (TUBES, POINTS), AFA (TUBES, POINTS)

DIMENSION AV (TUBES, POINTS), CV (TUBES, POINTS)

DIMENSION EV (TUBES, POINTS), GV (TUBES, POINTS), F (TUBES, POINTS)

DIMENSION LLINK (TUBES), RLINK (TUBES), TLINK (TUBES), PLINK (TUBES)

DIMENSION D (TUBES), XLTERM (TUBES), DMTERM (TUBES), QSTEDY (TUBES)

DIMENSION PAMP (FORCP), NTPROF (20), LPROF (20), NPTSVE (TERMS)

DIMENSION NTSTEN (10), PSTEN (10), CCV (TUBES), QGNRIC (TUBES)

DIMENSION NTANUR (10), PANUR (10), PALFA (10), LINEAR (TUBES), DIASAV (10)

DIMENSION RTUBE (TERMS), DIA (TUBES, POINTS), QS (TERMS), IRTOT (TERMS)

DIMENSION FLAG (TUBES), ISIGN (TUBES, 4), LQJ (TUBES, 4), NTJ (TUBES),

\*LPTB (TUBES, 4), KJ (TUBES, 4), LPJ (TUBES, 4), NTSP (TUBES), DEND (TUBES)

DIMENSION PMM (TUBES, POINTS), VP (30), VCLPLT (1, 1, 1), MPTSVE (TERMS)

DIMENSION NLines (10), PLOT (OTHER), PQ (OTHER), PR (OTHER), QCCPM (TUBES)

DIMENSION QAVE (TUBES), PAVE (TUBES), FCQS (60), FCQC (60), DTPREND (TUBES)

DIMENSION NODSTEN (10), KALF1 (TUBES), AA (POINTS), CCTERM (TERMS)

DIMENSION FCPS (60), FCPC (60), ZF (60), PHZP (60), PHZQ (60), DX (TUBES)

DIMENSION CCYCTM (OTHER), PP (NPS, OTHER), QQ (NPS, OTHER), PCH (TERMS)

DIMENSION RPLT (OTHER), VCPLT (OTHER), MLINK (TUBES), DTAPER (TUBES)

DIMENSION ALFG (30), KRT1 (TUBES), KDI1 (TUBES), NOSEGS (10), PC (TERMS)

DIMENSION KSOURC (TUBES), QLINK (TUBES), QOPP (TUBES), QMEAS (TUBES)

DIMENSION Pa11 (TUBES, POINTS), Qa11 (TUBES, POINTS), QGOAL (TUBES)

DIMENSION QMOS (TUBES), SSUM (TUBES), QDIFF (TUBES), XLINK (TUBES)

DIMENSION XLQTODL (TUBES), QMOQA (TUBES), QAVG (TUBES), ZLINK (TUBES)

C

```

CHARACTER*1 MENO1
CHARACTER*11 XYZ,ABCDE
CHARACTER*12 MENO,DNU
WRITE(6,174) TUBES,POINTS,TERMS
174 FORMAT('  PROGRAM CIRC  PARAMETER VALUES:  TUBES=',
* I3,' POINTS=',I2,' TERMINALS=',I2)
OPEN(UNIT=3,FILE='WR2',STATUS='UNKNOWN',FORM='FORMATTED')
88882 OPEN(UNIT=4,FILE='SSOS.DAT',STATUS='OLD')
OPEN(UNIT=20,FILE='CHI',STATUS='UNKNOWN',FORM='FORMATTED')
88883 OPEN(UNIT=8,FILE='WR28',STATUS='UNKNOWN',FORM='FORMATTED')
OPEN(UNIT=7,FILE='WR8',STATUS='UNKNOWN',FORM='FORMATTED')
OPEN(UNIT=13,FILE='QMEAS.DAT',STATUS='UNKNOWN',
*FORM='FORMATTED')
OPEN(UNIT=23,FILE='QGNRIC.DAT',STATUS='UNKNOWN',
*FORM='FORMATTED')
IMEIDE = 14
OPEN(UNIT=IMEIDE,FILE='MEIDE',STATUS='UNKNOWN',FORM='FORMATTED')
CALL TIME(ABCDE)
WRITE(3,13131) ABCDE
13131 FORMAT('TIME ABCDE =',A11)
DO 22222 I=1,TUBES
LP(I)=0
LQ(I)=0
IF(I.GT.10) GOTO 22222
NTSTEN(I)=0
NODSTEN(I)=0
PSTEN(I)=0.
NOSEGS(I)=0
NTANUR(I)=0
PANUR(I)=0.
PALFA(I)=0.
22222 CONTINUE
PI=3.141592654
PSAV22=0.
PSAV28=0.
PSAV42=0.
PSAV48=0.
QSAV22=0.
QSAV28=0.
QSAV42=0.
QSAV48=0.
C FOR REFLECTION COEFFICIENT DATA, SET IRC=1 AND NOHRMS=5, SAY
C FOR CALCULATED WAVE SPEED IN SELECTED VESSELS, SET IWS=1
C TO CONTINUE PREVIOUS JOB, SET ICONTIN=1 FOR STMT 1101 DO-LOOP
C NTUBES-NUMBER OF TUBES
C NTERM-NUMBER OF TERMINATIONS
READ(4,100) NTUBES,NTERM,ICONTIN,IRC,NOHRMS,IWS,NST,NTS
1,KCAMX,NSTA,NSTB,NTSA,NTSB,KTSB,ITEST
IF(KCAMX.LT.500) WRITE(3,127)
IF(KCAMX.LT.500) WRITE(6,127)
IF(KTSB.LE.0) KTSB=1
READ(4,100) (IRTOT(JJ),JJ=1,NTERM)
C ISOURCE=0 FOR FLOW SOURCE - OTHERWISE PRESSURE SOURCE
READ(4,100) IBINARY,ISOURCE,ITAPER,IZC,II1,JJ1,JJ1A,II2,JJ2,JJ2A
WRITE(6,101) NTUBES,NTERM,ICONTIN,IRC,NOHRMS,IWS,KCAMX
1,NST,NSTA,NSTB,NTS,NTSA,NTSB,KTSB,ITEST
WRITE(3,101) NTUBES,NTERM,ICONTIN,IRC,NOHRMS,IWS,KCAMX

```

```

1,NST,NSTA,NSTB,NTS,NTSA,NTSB,KTSB,ITEST
WRITE(6,100) IBINARY,ISOURCE,ITAPER,IZC,II1,JJ1,JJ1A,II2,JJ2,JJ2A
WRITE(3,100) IBINARY,ISOURCE,ITAPER,IZC,II1,JJ1,JJ1A,II2,JJ2,JJ2A
WRITE(6,100) (IRTOT(LL),LL=1,NTERM)
WRITE(3,100) (IRTOT(LL),LL=1,NTERM)
C IRC=0,SKIP RC CALCS; =1 NO SKIP. NOHRMS=NO.HARMONICS FOR RC
C ESTABLISH THE MAP
C LLINK-POINTS TO LEFT(OR ONLY) EXIT TUBE NUMBER
C RLINK-POINTS TO RIGHT EXIT TUBE NUMBER(IF NEEDED)
C TLINK-POINTS TO TERMINAL TUBE ADDITIONAL INFORMATION
C PLINK-POINTS TO PRESSURE INPUT TUBES
C QLINK-POINTS TO FLOW INPUT TUBES
C TQLINK-POINTS TO TERMINAL FLOW TUBES
C ZLINK-POINTS TO ZAG-LIKE TUBES WHERE RCR IS TO BE DETERMINED
    READ(13,15951) (QMEAS(I),I=1,NTUBES)
    WRITE(6,15952) (QMEAS(I),I=1,NTUBES)
15952 FORMAT(2X,'QMEAS(I)=' ,8F7.2)
15951 FORMAT(10F10.4)
    READ(23,15951) (QGNRIC(I),I=1,NTUBES)
    WRITE(6,15953) (QGNRIC(I),I=1,NTUBES)
15953 FORMAT(2X,'QGNRIC(I)=' ,8F7.2)
    READ(4,10011) (LLINK(I),I=1,NTUBES)
10011 FORMAT(15I4)
    WRITE(6,102) (LLINK(I),I=1,NTUBES)
    WRITE(3,102) (LLINK(I),I=1,NTUBES)
    READ(4,10011) (RLINK(I),I=1,NTUBES)
    WRITE(6,104) (RLINK(I),I=1,NTUBES)
    WRITE(3,104) (RLINK(I),I=1,NTUBES)
    READ(4,10011) (MLINK(I),I=1,NTUBES)
    WRITE(6,105) (MLINK(I),I=1,NTUBES)
    WRITE(3,105) (MLINK(I),I=1,NTUBES)
C FOR ANEURYSM VESSELS, MAKE Q AT END EQUAL ZERO (STMT IN 7090 DO-LOOP)
C ALSO SET TLINK TO ZERO FOR ANEURYSM VESSELS
    READ(4,100) (TLINK(I),I=1,NTUBES)
    WRITE(6,106) (TLINK(I),I=1,NTUBES)
    WRITE(3,106) (TLINK(I),I=1,NTUBES)
    READ(4,100) (PLINK(I),I=1,NTUBES)
    WRITE(6,108) (PLINK(I),I=1,NTUBES)
    WRITE(3,108) (PLINK(I),I=1,NTUBES)
    READ(4,100) (QLINK(I),I=1,NTUBES)
    WRITE(6,109) (QLINK(I),I=1,NTUBES)
    WRITE(3,109) (QLINK(I),I=1,NTUBES)
    READ(4,100) (TQLINK(I),I=1,NTUBES)
    WRITE(6,103) (TQLINK(I),I=1,NTUBES)
    WRITE(3,103) (TQLINK(I),I=1,NTUBES)
    READ(4,100) (ZLINK(I),I=1,NTUBES)
    WRITE(6,11103) (ZLINK(I),I=1,NTUBES)
    WRITE(3,11103) (ZLINK(I),I=1,NTUBES)
C LP-NUMBER OF POINTS WHERE PRESSURE IS CALCULATED, STARTING WITH AND
C ENDING WITH THE CENTER POINT OF A JUNCTION
C D-DIAMETER OF TUBE AT NODE 1, DIA-DIAMETER OF TUBE AT EACH NODE
C ALFA-NOMINAL TUBE STIFFNESS FACTOR
C XLTERM-LENGTH OF TERMINAL TUBE(STEADY STATE)
C DMTERM-DIAMETER OF TERMINAL TUBE(STEADY STATE)
C QSTEDY-STEADY FLOW VALUES AT TERMINATIONS
    READ(4,100) (LP(I),I=1,NTUBES)
    WRITE(3,100) (LP(I),I=1,NTUBES)

```

```

DO 11098 I=1,NTUBES
11098 READ(4,11099) JUNK,D(I),DTAPER(I)
11099 FORMAT(I5,2F10.5)
DO I=1,NTUBES
WRITE(3,11099) I,D(I),DTAPER(I)
END DO
91001 READ(4,110) (ALFG(K),K=1,30)
READ(4,110) ALFGFAC
DO 98551 I=1,30
ALFG(I)=ALFG(I)*ALFGFAC
98551 CONTINUE
READ(4,110) DXTD,PZC,PSTOP,PINITAL,QINITAL,PQMOD,DFUNST
PINI=PINITAL/1333.
WRITE(6,110) DXTD,PZC,PSTOP,PINI,QINITAL,PQMOD,DFUNST
WRITE(3,110) DXTD,PZC,PSTOP,PINI,QINITAL,PQMOD,DFUNST
DO 91055 I=1,NTUBES
D(I)=D(I)*DFUNST
91055 DTAPER(I)=DTAPER(I)*DFUNST
PSTOP=PSTOP*1333.
CWHEN TAPERING, USE AXTD FOR FINAL AREA
C AXTD=3.14159*DXTD*DXTD/4.
READ(4,128) NALF1,ALFA,ALFA1,PFCTOR,DFCTOR,NDI1,RFCTOR,NRT1
C IF(PFCTOR.LT.0.1.OR.PFCTOR.GT.2.0) PFCTOR=1.
WRITE(6,125) NALF1,ALFA,ALFA1,PFCTOR,DFCTOR,NDI1,RFCTOR,NRT1
WRITE(3,125) NALF1,ALFA,ALFA1,PFCTOR,DFCTOR,NDI1,RFCTOR,NRT1
C IF(PFCTOR.LT.0.1.OR.PFCTOR.GT.2.5) PFCTOR=1.
C IF(DFCTOR.LT.0.1.OR.DFCTOR.GT.2.) DFCTOR=1.
C IF(RFCTOR.LT.0.01.OR.RFCTOR.GT.100.) RFCTOR=1.
IF(NALF1.GE.1) READ(4,100) (KALF1(I),I=1,NALF1)
IF(NALF1.GE.1) WRITE(6,126) (KALF1(I),I=1,NALF1)
IF(NALF1.GE.1) WRITE(3,126) (KALF1(I),I=1,NALF1)
IF(NDI1.GE.1) READ(4,100) (KDI1(I),I=1,NDI1)
IF(NDI1.GE.1) WRITE(6,99126) (KDI1(I),I=1,NDI1)
IF(NDI1.GE.1) WRITE(3,99126) (KDI1(I),I=1,NDI1)
125 FORMAT(2X,'*** ALFA',I9,2F5.1,' P.FCT',F5.3,' D.FCT',F5.3,I5,
*' R.FCT',F5.3,I5)
126 FORMAT(2X,'ALFA1 VESSELS ',20I3)
98126 FORMAT(2X,'TERM1 VESSELS ',20I3)
99126 FORMAT(2X,'DIAM1 VESSELS ',20I3)
127 FORMAT(' NO.OF LINES STORED FOR P-Q-A FILE LESS THAN 500')
128 FORMAT(I5,4F5.1,I5,F5.1,I5)
IF(NDI1.LE.0) GOTO 91004
DO 91002 I=1,NDI1
IF(ITAPER.NE.0)DTAPER(KDI1(I))=DTAPER(KDI1(I))*DFCTOR
IF(ITAPER.NE.0)DTAPER(KDI1(I))=DTAPER(KDI1(I))*DFCTOR
91002 D(KDI1(I))=D(KDI1(I))*DFCTOR
91004 READ(4,130) (XLTERM(I),I=1,NTERM)
READ(4,110) (DMTERM(I),I=1,NTERM)
DO 99105 I=1,NTERM
99105 DMTERM(I)=DMTERM(I)*DFCTOR
99104 READ(4,110) (QSTEDY(I),I=1,NTERM)
READ(4,110) (CCTERM(I),I=1,NTERM)
IF(NRT1.LE.0) GOTO 91006
READ(4,100) (KRT1(I),I=1,NRT1)
WRITE(6,98126) (KRT1(I),I=1,NRT1)
WRITE(3,98126) (KRT1(I),I=1,NRT1)
DO 91005 I=1,NRT1

```

```

91005 XLTERM(KRT1(I))=XLTERM(KRT1(I))*RFCTOR
91006 DO 401 I=1,NTERM
      XLTERM(I)=XLTERM(I)*PFCTOR
C      IF(CCTERM(I).LT.0.001) CCTERM(I)=1.
      401 CONTINUE
C QSTEDY CAN BE IN ML/MIN OR CC/SEC SINCE ONLY THEIR RATIOS ARE USED
      WRITE(6,400) (D(I),I=1,NTUBES)
      WRITE(3,400) (D(I),I=1,NTUBES)
      WRITE(6,99400) (DTAPER(I),I=1,NTUBES)
      WRITE(3,99400) (DTAPER(I),I=1,NTUBES)
      WRITE(6,410) (XLTERM(I),I=1,NTERM)
      WRITE(3,410) (XLTERM(I),I=1,NTERM)
      WRITE(6,420) (DMTERM(I),I=1,NTERM)
      WRITE(3,420) (DMTERM(I),I=1,NTERM)
      WRITE(6,411) (CCTERM(I),I=1,NTERM)
C
C READ FILE POINTERS IP,IQ,IA,IIM
      READ(4,100) IP,IQ,IIM,IA
      IF(IP.GE.1) IP=10
      IF(IQ.GE.1) IQ=11
      IF(IMM.GE.1) IMM=12
      IF(IA.GE.1) IA=9
C
C NPRIPP-NUMBER OF TIMES SOLUTION PRINTED PER PERIOD
C NPERM-NUMBER OF PERIODS, MAXIMUM
C NPERL-NUMBER OF PERIODS, LINEAR
C TIMP-NUMBER OF TIMES IMPEDENCE PRESSURES AND FLOWS SAVED
C NTDIV-NUMBER OF TIME DIVISIONS PER PERIOD
      READ(4,140) NPERM,NPERL,NPERP,NPRIPP,NTDIV,NTDIV2,NTDIV3,
      *NTDIV4,TIMP
      IF(ICONTIN.NE.0) NPERL=0
      READ(4,150) XRTOT,PV,PO,RHO,XMU,XMUSTR,PRESI
      READ(4,160) HR,DX1,DX2,PM,DELP,CVTOT
C CVTOT READ IN IS IN CC/MM HG, CONVERT NOW TO CC/DYNES/SQCM
      CVTOT=CVTOT/1333.
      READ(4,110) (DX(N),N=1,NTUBES)
C      IF(DX1.GT.0.0001) THEN
C      DO 11099 N=1,NTUBES
C11099 DX(N)=DX1
C      IF(NST.GT.0) DX(NST)=DX2
C      IF(NTS.GT.0) DX(NTS)=DX2
C      ENDIF
C
C DETERMINE WHETHER SINE FORCING FUNCTION USED
C IF NSINUO=0,USE FUGEN (FUNCTION GENERATOR)
      READ(4,100) NSINUO,IFORIER
      IF (NSINUO.NE.0) THEN
      READ(4,130) (PAMP(I),I=1,NSINUO)
      WRITE(3,411) (CCTERM(I),I=1,NTERM)
      WRITE(6,135) (PAMP(I),I=1,NSINUO)
      WRITE(3,135) (PAMP(I),I=1,NSINUO)
      135 FORMAT(2X,'PAMPS FOLLOW',8F10.1)
      ENDIF
C
C VELOCITY PROFILE PLOTTING
C NPROFL-NUMBER OF VELOCITY PROFILES TO BE PLOTTED
C NTPROF-TUBE NUMBERS FOR WHICH VELOCITY PROFILES ARE TO BE PLOTTED

```

```

      READ(4,100) NPROFL
      WRITE(6,10099) NSINUO,IFORIER,NPROFL
      WRITE(3,10099) NSINUO,IFORIER,NPROFL
10099  FORMAT(2X,'NSINUO',I3,2X,'IFORIER',I3,2X,'VEL PROF PLTTED',I5)
      IF (NPROFL.NE.0) THEN
      READ(4,100) (NTPROF(I),I=1,NPROFL)
      READ(4,100) (LPROF(I),I=1,NPROFL)
      WRITE(6,440) (NTPROF(I),I=1,NPROFL)
      WRITE(3,440) (NTPROF(I),I=1,NPROFL)
      ENDIF
C
C IMPEDENCE CALCULATIONS
C NPSAVE=NUMBER OF TUBES WHERE INPUT IMPEDENCE CALCULATED
C NPTSVE=TUBE NUMBERS WHERE INPUT IMPEDENCE CALCULATED
      READ(4,100) NPSAVE
      IF (NPSAVE.NE.0) THEN
      READ(4,100) (NPTSVE(I),I=1,NPSAVE)
      READ(4,100) (MPTSVE(I),I=1,NPSAVE)
      WRITE(6,460) NPSAVE,(NPTSVE(I),I=1,NPSAVE)
      WRITE(3,460) NPSAVE,(NPTSVE(I),I=1,NPSAVE)
      WRITE(6,46099) (MPTSVE(I),I=1,NPSAVE)
      WRITE(3,46099) (MPTSVE(I),I=1,NPSAVE)
      ENDIF
46099  FORMAT(' ',' ' NODAL PTS SPEC CALCS IIM FILE 80F',15I4)
C STENOSIS CALCULATIONS
C ISTE=NUMBER OF STENOSES, IANUR=NUMBER OF ANEURYSMS
C NTSTEN,NTANEU=TUBE NUMBERS FOR STENOSES AND ANEURYSMS
C PSTEN,PANUR,PALFA=SEVERITY FACTORS FOR STEN,ANUR,WALL FLEX
      READ(4,100) ISTE,IANUR,JANUR,KANUR
      IF (JANUR.GT.0) IANUR=0
C BOTH JANUR & KANUR MUST BE > 0 FOR USING TER.RES.FOR ANEURISM
      WRITE(6,10098) ISTE,IANUR,JANUR,KANUR
      WRITE(3,10098) ISTE,IANUR,JANUR,KANUR
10098  FORMAT(' ISTE=',I3,2X,'IANUR=',I3,2X,'JANUR&KANUR=',2I3)
      DO 67823 K=1,NTUBES
67823  LINEAR(K)=0
      IF (ISTE.NE.0) THEN
      READ(4,100) (NTSTEN(I),I=1,ISTE)
C      NST=NTSTEN(1) CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 98723 I=1,ISTE
98723  LINEAR(NTSTEN(I))=1
      READ(4,110) (PSTEN(I),I=1,ISTE)
      READ(4,100) (NODSTEN(I),I=1,ISTE)
      READ(4,100) (NOSEGS(I),I=1,ISTE)
      WRITE(6,470) (NTSTEN(I),I=1,ISTE)
      WRITE(3,470) (NTSTEN(I),I=1,ISTE)
      WRITE(6,480) (PSTEN(I),I=1,ISTE)
      WRITE(3,480) (PSTEN(I),I=1,ISTE)
      WRITE(6,9470) (NODSTEN(I),I=1,ISTE)
      WRITE(3,9470) (NODSTEN(I),I=1,ISTE)
      WRITE(6,8470) (NOSEGS(I),I=1,ISTE)
8470  FORMAT(2X,'NOSEGS',5I5)
      ENDIF
C BEST NOT TO USE IANUR=1 WITH PANUR AND PALFA=0 TO ELIMINATE ANEURYSM CALCS
      IF (IANUR.NE.0) THEN
      READ(4,100) (NTANUR(I),I=1,IANUR)
      READ(4,110) (PANUR(I),I=1,IANUR)

```

```

      READ(4,110) (PALFA(I),I=1,IANUR)
      WRITE(3,8470) (NOSEGS(I),I=1,ISTEN)
      WRITE(6,99470) (NTANUR(I),I=1,IANUR)
      WRITE(3,99470) (NTANUR(I),I=1,IANUR)
      WRITE(6,481) (PANUR(I),I=1,IANUR)
      WRITE(3,481) (PANUR(I),I=1,IANUR)
      WRITE(6,482) (PALFA(I),I=1,IANUR)
      WRITE(3,482) (PALFA(I),I=1,IANUR)
      ENDIF
      IF(JANUR.GT.0) THEN
        READ(4,100) (NTANUR(I),I=1,JANUR)
        READ(4,110) (PANUR(I),I=1,JANUR)
        READ(4,110) (PALFA(I),I=1,JANUR)
        WRITE(6,99470) (NTANUR(I),I=1,IANUR)
        WRITE(3,99470) (NTANUR(I),I=1,IANUR)
        WRITE(6,482) (PALFA(I),I=1,JANUR)
        WRITE(3,482) (PALFA(I),I=1,JANUR)
      ENDIF
C NFORCP-NUMBER OF TUBES WITH PRESSURE FORCING FUNCTIONS AT INLETS
C NFORCQ-NUMBER OF TUBES WITH FLOW FORCING FUNCTIONS AT INLETS
C NFORCT-NUMBER OF TUBES WITH TERMINAL FORCING FUNCTIONS
C PMULT,QMULT,TMULT-MULTIPLYING FACTOR FOR INPUT SIGNAL
C TLAGP,TLAGQ,TLAGT-PHASE LAG FOR INPUT SIGNAL
C IFRIC-WALL SHEAR COMPUTATION FLAG
C NPTSFFP,NPTSFFQ,NPTSFFT-NUMBER OF POINTS IN FORCING FUNCTION
      READ(4,100) NPTSFFP,NFORCP,NPTSFFQ,NFORCQ,NPTSFFT,NFORCT
      IF(NFORCP.NE.0) THEN
        READ(4,910) (PMULT(I),I=1,NFORCP)
        READ(4,910) (TLAGP(I),I=1,NFORCP)
        WRITE(6,91477)
        WRITE(6,91470) (PMULT(I),I=1,NFORCP)
        WRITE(6,91470) (TLAGP(I),I=1,NFORCP)
        WRITE(3,91477)
        WRITE(3,91470) (PMULT(I),I=1,NFORCP)
        WRITE(3,91470) (TLAGP(I),I=1,NFORCP)
      END IF
      IF(NFORCQ.NE.0) THEN
        READ(4,910) (QMULT(I),I=1,NFORCQ)
        READ(4,910) (TLAGQ(I),I=1,NFORCQ)
        WRITE(6,91471)
        WRITE(6,91470) (QMULT(I),I=1,NFORCQ)
        WRITE(6,91470) (TLAGQ(I),I=1,NFORCQ)
        WRITE(3,91471)
        WRITE(3,91470) (QMULT(I),I=1,NFORCQ)
        WRITE(3,91470) (TLAGQ(I),I=1,NFORCQ)
      END IF
      IF(NFORCT.NE.0) THEN
        READ(4,910) (TMULT(I),I=1,NFORCT)
        READ(4,910) (TLAGT(I),I=1,NFORCT)
        WRITE(6,91472)
        WRITE(6,91470) (TMULT(I),I=1,NFORCT)
        WRITE(6,91470) (TLAGT(I),I=1,NFORCT)
        WRITE(3,91472)
        WRITE(3,91470) (TMULT(I),I=1,NFORCT)
        WRITE(3,91470) (TLAGT(I),I=1,NFORCT)
      END IF
910 FORMAT(10F8.5)

```



```

91477 FORMAT(2X, 'PMULT & TLAGP FOR INLET PRESS FORC FUNC')
91470 FORMAT(2X, 10F8.5)
91471 FORMAT(2X, 'QMULT & TLAGQ FOR INLET FLOW FORC FUNC')
91472 FORMAT(2X, 'TMULT & TLAGT FOR TERM FLOW FORC FUNC')
  READ(4,100) IFRIC
  NTSPEC=0
  DO I=1,NTUBES
    IF (TLINK(I).NE.0.AND.D(I).LT.0.15) THEN
      NTSPEC=NTSPEC+1
      NTSP(NTSPEC)=I
      DEND(I)=D(I)
      DTPREND(I)=DTAPER(I)
      D(I)=0.15
      DTAPER(I)=0.15
    END IF
  END DO
66669 WRITE(6,560) IFRIC
  WRITE(3,560) IFRIC
  IF(NTSPEC.EQ.0) GO TO 66668
  WRITE(3,66710) NTSPEC
66710 FORMAT(' NTSPEC = ',I5)
  WRITE(3,66707) (NTSP(I),I=1,NTSPEC)
66707 FORMAT(' NTSP : ',5I3)
  WRITE(3,66708) (DEND(NTSP(I)),I=1,NTSPEC)
66708 FORMAT(' DEND : ',5F10.5)
  WRITE(3,66709) (DTPREND(NTSP(I)),I=1,NTSPEC)
66709 FORMAT(' DTPREND : ',5F10.5)
C PFG,QFG,TFG-DIGITAL BREAKUP OF PHYSIOLOGICAL FORCING FUNCTIONS
66668 IF(NFORCP.NE.0) THEN
  PXMA=0.
  PXMI=500.
C   IF (NSINUO.EQ.0) THEN
  DO 89898 K=1,NFORCP
    READ(4,98110) (PFG(K,I),I=1,NPTSFFP)
89898 CONTINUE
98110 FORMAT (8F10.5)
    DO 44402 K=1,NFORCP
      DO 402 I=1,NPTSFFP
        IF(PXMA.LT.PFG(K,I)) PXMA=PFG(K,I)
        IF(PXMI.GT.PFG(K,I)) PXMI=PFG(K,I)
      402 PFG(K,I)=PFG(K,I)+PQMOD
    C402 PFG(K,I)=PFG(K,I)*PFCTOR
  44402 CONTINUE
    WRITE(3,9875)NFORCP,NPTSFFP
9875  FORMAT('  NFORCP  ',I3,'  NPTSFFP  ',I3)
    DO 89897 K=1,NFORCP
      WRITE(6,110) (PFG(K,I),I=1,NPTSFFP)
      WRITE(3,110) (PFG(K,I),I=1,NPTSFFP)
89897 CONTINUE
    END IF
C
  IF(NFORCT.NE.0) THEN
    TXMA=0.
    TXMI=500.
C   IF (NSINUO.EQ.0) THEN
  DO 59898 K=1,NFORCT
    READ(4,98110) (TFG(K,I),I=1,NPTSFFT)

```

```

59898 CONTINUE
      DO 54402 K=1,NFORCT
      DO 502 I=1,NPTSFFT
      IF (TXMA.LT.TFG(K,I)) TXMA=TFG(K,I)
      IF (TXMI.GT.TFG(K,I)) TXMI=TFG(K,I)
502   TFG(K,I)=TFG(K,I)+PQMOD
C502   TFG(K,I)=TFG(K,I)*TFCTOR
54402 CONTINUE
      WRITE(3,59875)NFORCT,NPTSFFT
59875 FORMAT('  NFORCT  ',I3,'  NPTSFFT  ',I3)
      DO 29897 K=1,NFORCT
      WRITE(6,110) (TFG(K,I),I=1,NPTSFFT)
      WRITE(3,110) (TFG(K,I),I=1,NPTSFFT)
29897 CONTINUE
      END IF
C
      IF(NFORCQ.NE.0) THEN
        QXMA=0.
        QXMI=500.
C
      IF (NSINUO.EQ.0) THEN
      DO 69898 K=1,NFORCQ
      READ(4,98110) (QFG(K,I),I=1,NPTSFFQ)
69898 CONTINUE
      DO 64402 K=1,NFORCQ
      DO 602 I=1,NPTSFFQ
      IF (QXMA.LT.QFG(K,I)) QXMA=QFG(K,I)
      IF (QXMI.GT.QFG(K,I)) QXMI=QFG(K,I)
602   QFG(K,I)=QFG(K,I)+PQMOD
C602   QFG(K,I)=QFG(K,I)*QFCTOR
64402 CONTINUE
      WRITE(3,79875)NFORCQ,NPTSFFQ
79875 FORMAT('  NFORCQ  ',I3,'  NPTSFFQ  ',I3)
      DO 69897 K=1,NFORCQ
      WRITE(6,110) (QFG(K,I),I=1,NPTSFFQ)
      WRITE(3,110) (QFG(K,I),I=1,NPTSFFQ)
69897 CONTINUE
      END IF
C
100  FORMAT(20I3)
101  FORMAT(2X,'TUBES=',I3,2X,'TERMS=',I3,2X,'CONTIN=',I3,
12X,'IRC=',I3,2X,'NOHRMS=',I3,2X,'IWS=',I3,2X,'KCAMX=',I3,2X/
12X,'NST=',3I4,2X,'NTS=',3I4)
11101 FORMAT(2X,'TAPER=',I3,2X,'WAVE SPEED VESSELS AND NODE SPAN',6I3)
102  FORMAT(2X,'LLINK(J)  FOLLOWS',15I4)
104  FORMAT(2X,'RLINK(J)  FOLLOWS',20I3)
105  FORMAT(2X,'MLINK(J)  FOLLOWS',15I4)
106  FORMAT(2X,'TLINK(J)  FOLLOWS',20I3)
108  FORMAT(2X,'PLINK(J)  FOLLOWS',20I3)
109  FORMAT(2X,'QLINK(J)  FOLLOWS',20I3)
103  FORMAT(2X,'TQLINK(J) FOLLOWS',20I3)
11103 FORMAT(2X,'ZLINK(J) FOLLOWS',20I3)
110  FORMAT(8F10.5)
120  FORMAT(26F3.1)
130  FORMAT(8F10.1)
140  FORMAT(9I5)
150  FORMAT(F10.4,2F10.1,3F10.4,1F10.1)
160  FORMAT(F10.1,2F10.3,2F10.1,F10.8)

```

```

C
  IF (ICONTIN.NE.0) THEN
    OPEN(UNIT=9, FILE='WRPOLD', STATUS='OLD', FORM='FORMATTED')
    OPEN(UNIT=10, FILE='WRQOLD', STATUS='OLD', FORM='FORMATTED')
    DO 33331 I=1, NPRIPP-1
      READ(9,170) CYCTM
      READ(10,170) CYCTM
    DO 33331 M=1, NTUBES
      READ(9,200) (PMM(M,K), K=1, LP(M))
      READ(10,99200) (Q(M,K), K=1, LQ(M))
33331 WRITE(6,33334) I,M,CYCTM
33334 FORMAT('+ OLD FILE', 2I5, F10.3)
      READ(9,170) CYCTM
      READ(10,170) CYCTM
      END IF
      MMCTRL=0
      MEECCC=0
91817 MMCTRL=MMCTRL+1
      DO 1101 I=1, NTUBES
        IF (ICONTIN.EQ.0) THEN
          LQ(I)=LP(I)
          IF (TLINK(I).EQ.0) LQ(I)=LP(I)-1
C LP=LQ FOR INLET TUBES FOR FLOW SOURCE
C IF (ISOURCE.EQ.0.AND.PLINK(I).NE.0) LQ(I)=LP(I)
          DO 1010 N=1, LQ(I)
            1010 Q(I,N)=QINITAL
            DO 1020 N=1, LP(I)
              PH(I,N)=PINITAL
              P(I,N)=PINITAL
C IF (PRESI.GT.PV) PH(I,J)=PRESI
C IF (PRESI.GT.PV) P(I,J)=PRESI
            1020 CONTINUE
            ELSE
C FIRST MANUFACTURE A READ FILE FROM A PREVIOUS RUN BY DELETING ALL
C DATA UP TO LAST TIME STEP THEN READ IT IN ON FILES 20 AND 40
              READ(9,200) (PMM(I,K), K=1, LP(I))
              DO 77096 K=1, LP(I)
27096 P(I,K)=PMM(I,K)*1333.2
              READ(10,99200) (Q(I,K), K=1, LQ(I))
              WRITE(6,33334) I, LQ(I), Q(1,1)
C DO 77097 K=1, LQ(I)
C77097 Q(I,K)=BB( K)
              ENDIF
            1101 CONTINUE
              IF (ICONTIN.NE.0) CLOSE(9)
              IF (ICONTIN.NE.0) CLOSE(10)
              NSTB1=NSTB
              NTSB1=NTSB
              IF (NST.LE.1.AND.LQ(NST).LT.NSTB1) NSTB1=LQ(NST)
              IF (NTS.GE.1.AND.LQ(NTS).LT.NTSB1) NTSB1=LQ(NTS)
              IF (IANUR.GE.1) THEN
                DO 81101 JJ=1, IANUR
                  LQ(NTANUR(JJ))=LP(NTANUR(JJ))
81101 CONTINUE
                ENDIF
C DO YOU WANT TO TAPER ANY VESSELS? TAPER=0,NO;=1,YES
              IF (ITAPER.EQ.0) GO TO 1001

```

```

DO 9830 KK=1,NTUBES
AO(KK,1)=(3.14159*D(KK)*D(KK))/4.
DIA(KK,1)=SQRT(4.*AO(KK,1)/3.14159)
L=LP(KK)
DIF=AO(KK,1)-DTAPER(KK)*DTAPER(KK)*3.14159/4.
DO 9829 J=2,L
AO(KK,J)=AO(KK,1)-(J-1)*DIF/L
DIA(KK,J)=SQRT(4.*AO(KK,J)/3.14159)
9829 CONTINUE
9830 CONTINUE
MEECC=1
IF(MEECC.EQ.1) GO TO 1001
DO 9930 KK=1,NTUBES
L=LP(KK)
KL=KK+1
IF(KK.EQ.NTUBES) DUMY=AXTD
IF(KK.NE.NTUBES) DUMY=AO(KL,1)
DIF=AO(KK,1)-DUMY
DO 9929 J=2,L
AO(KK,J)=AO(KK,1)-(J-1)*DIF/L
9929 CONTINUE
9930 CONTINUE
23131 DO 9932 KF=1,NTUBES
L=LP(KF)
9932 WRITE(6,9933) (AO(KF,J),J=1,L)
WRITE(3,9933) (AO(KF,J),J=1,L)
9933 FORMAT(2X,10F8.5)
C LINEARLY INTERPOLATE TO GET ELASTIC TAPER FOR ALL VESSELS
C ALFA IS CENTRAL VALUE IN INTERPOLATION TABLE ALFG
C D3 IS LARGEST DIAM IN MODEL, NOMINAL ALFA IS 4.0
C BUT CAN BE READ IN AT ANY VALUE DESIRED
1001 D3=3.
DO J=1,NTUBES
IF(D3.LT.D(I)) D3=D(I)
IF(D3.LT.DTAPER(I)) D3=DTAPER(I)
END DO
DO 97071 J=1,NTUBES
LL=LP(J)
ZKK=ALFA/4.
IF(NALF1.LE.0) GOTO 97073
DO 97072 L=1,NALF1
IF(J.NE.KALF1(L)) GOTO 97072
ZKK=ALFA1/4.
GOTO 97073
97072 CONTINUE
97073 IF(KANUR.LE.0.OR.JANUR.LE.0) GOTO 97075
DO 97074 L=1,JANUR
IF(J.NE.NTANUR(L)) GOTO 97074
ZKK=PALFA(L)/4.
GOTO 97075
97074 CONTINUE
97075 DO 97070 L=1,LL
ZK=DIA(J,L)*29./D3
K=ZK+1
IF(K.GE.30) THEN
AFA(J,L)=ALFG(30)*ZKK
ELSE

```

CANVAS

END IF

97070 CONTINUE

C AFA(J,1)=AFA(J,1)/3.

C AFA(J,LL)=AFA(J,LL)/3.

97071 CONTINUE

IF(ISTEN.EQ.0) GO TO 87074

KZ=NODSTEN(1)

C AFA(2,KZ)=2.

C AFA(2,KZ+1)=2.

C AFA(2,KZ+2)=4.

C AFA(2,KZ-1)=4.

C WRITE(6,9933) AFA(2,KZ-2),AFA(2,KZ-1),AFA(2,KZ),AFA(2,KZ+1)

C 1,AFA(2,KZ+2),AFA(2,KZ+3)

C WRITE(3,9933) AFA(2,KZ-2),AFA(2,KZ-1),AFA(2,KZ),AFA(2,KZ+1)

C 1,AFA(2,KZ+2),AFA(2,KZ+3)

C

C TO DETERMINE THE CAPACITANCE VALUE AT THE TERMINAL VESSELS:

C SUBTRACT THE CAPACITANCE OF ALL THE TUBES USED IN THE MODEL

C (SUM) FROM THE TOTAL (ESTIMATED) CAPACITANCE OF THE SYSTEM (CVTOT)

C TO GET THE REMAINING CAPACITANCE TO BE PLACED AT THE TERMS (CVTERM).

C DIVIDE THE REMAINING CAPACITANCE AMONG THE C'S IN THE RCR'S

C ACCORDING TO THE AMOUNT OF FLOW THROUGH THAT TERMINATION.

C QSSUM-TOTAL FLOW TO TERMINATIONS

C QS-PROPORTIONAL AMOUNT OF FLOW TO EACH TERMINATION

C CALCULATE FOR ALL TUBES THE FOLLOWING

C LQ-NUMBER OF POINTS WHERE FLOW IS CALCULATED. FOR TUBES WHICH ARE

C TERMINATED BY A JUNCTION: LQ=LP-1, OTHERWISE LQ=LP

C P-PRESSURE ARRAY(INITIALIZE)

C Q-FLOW ARRAY(INITIALIZE)

C AO-ORIGINAL TUBE CROSS SECTIONAL AREA

C SUM-TOTAL CAPACITANCE OF ALL VESSELS(UNSTEADY FLOW)

87074 SUMTOT=0.

SUM=0.

ABC=(PO+DELP/2.)/(PO-DELP/2.)

ALN=100.\*LOG(ABC)

PRINT \*, 'ABC,ALN FOLLOW',ABC,ALN

DO 1000 I=1,NTUBES

IF(TLINK(I).EQ.0) GO TO 10111

C DETERMINE THE TERMINAL RESISTANCE,RTOT

C RTUBE-RESISTANCE OF EFFERENT TUBE(UNSTEADY FLOW)

C RTOT-RESISTANCE OF EFFERENT TUBE(STEADY FLOW)-RTUBE

RTUBE(TLINK(I))=128.\*XMU\*DX(I)\*LP(I)/(3.1416\*D(I)\*\*4)

RTOT(TLINK(I))=128.\*XMU\*XLTERM(TLINK(I))/(3.1416\*

\*DMTERM(TLINK(I))\*\*4)-RTUBE(TLINK(I))

RESUTO=1./RTOT(TLINK(I))

SUMTOT=SUMTOT+RESUTO

C PRINT \*,I,TLINK(I),RTOT(TLINK(I)),RESUTO

WRITE(6,11991) I,RTUBE(TLINK(I)),DX(I),LP(I),D(I)

WRITE(6,11991) I,RTOT(TLINK(I)),XLTERM(TLINK(I)),DMTERM(TLINK(I))

WRITE(3,11991) I,RTUBE(TLINK(I)),DX(I),LP(I),D(I)

WRITE(3,11991) I,RTOT(TLINK(I)),XLTERM(TLINK(I)),DMTERM(TLINK(I))

11991 FORMAT(2X,I5,4F15.5)

LQ(I)=LP(I)

GO TO 10112

10111 LQ(I)=LP(I)-1

IF(TQLINK(I).NE.0) LQ(I)=LP(I)

```

10112      CONTINUE
C FOLLOWING 'IF' CAN IMPOSE ZC ON ANY TERMINATION
      IF(IRTOT(TLINK(I)).EQ.1) THEN
CANVAS
CANVAS
      ENDIF
      DO 91021 JJ=1, IANUR
      IF(I.EQ.NTANUR(JJ)) LQ(I)=LP(I)
91021 CONTINUE
      CCV(I)=(AFA(I,1)*AO(I,1)*LQ(I)*DX(I))/(PM*ALN)
      SUM=SUM+CCV(I)
1000 CONTINUE
      RTOTSUM=1./SUMTOT
      PRINT *,RTOTSUM
      CVTERM=CVTOT-SUM
      QSSUM=0.
      QSTOT=0.
      DO 2000 I=1, NTERM
2000 QSSUM=QSSUM+QSTEDY(I)
      DO 3000 I=1, NTERM
      QS(I)=QSTEDY(I)/QSSUM
3000 QSTOT=QSTOT+QS(I)
      DO 4000 I=1, NTERM
4000 CVTER(I)=(QS(I)/QSTOT)*CVTERM*CCTERM(I)
C
C FLAG FOR THE JUNCTION TUBES
C IF TUBES COME TOGETHER TO FORM A JUNCTION, FLAG = 1
C DON'T WANT TO DO A JUNC TWICE, SO SET FLAGS AHEAD IN SEQUENCE
C BUT NOT BACKWARDS
      DO 5000 J=1, NTUBES
5000 FLAG(J)=1
      DO 6000 J=1, NTUBES
      IF (LLINK(J).LE.0) THEN
C IF TERMINAL TUBE, FLAG = 0
      IF (LLINK(J).EQ.0) FLAG(J)=0
      IF (LLINK(J).LT.0) THEN
      LLJ=ABS(LLINK(J))
      LLLJ=ABS(LLINK(RLINK(J)))
CANVAS
CANVAS
CANVAS
CANVAS
      ENDIF
      ELSE
      LLO=ABS(LLINK(J))
      IF (RLINK(LLO).EQ.J.AND.LLO.GT.J) FLAG(LLO)=0
C 1ST TERM OF IF CHECKS IF HEADING TOWARDS SAME JUNC, 2ND IF BEEN THERE BEFORE
      ENDIF
      IF (RLINK(J).NE.0.AND.FLAG(J).NE.0) THEN
      LLLI=ABS(LLINK(RLINK(J)))
      IF (LLLI.EQ.J.AND.RLINK(J).GT.J) FLAG(RLINK(J))=0
      ENDIF
6000 CONTINUE
      WRITE(6,565) (FLAG(J),J=1,NTUBES)
      WRITE(3,565) (FLAG(J),J=1,NTUBES)
      TMAX=60./HR
      DT=TMAX/NTDIV

```

```

TP=TMAX/NPRIPP
TI=TMAX/TIMP
WRITE(6,93110) TMAX,DT,TP,TI
WRITE(3,93110) TMAX,DT,TP,TI
93110 FORMAT(2X,'TMAX,DT,TP,TI FOLLOW',8F10.5)
WRITE(6,490) (QSTEDY(I),I=1,NTERM)
WRITE(3,490) (QSTEDY(I),I=1,NTERM)
WRITE(6,500) (QS(I),I=1,NTERM)
WRITE(3,500) (QS(I),I=1,NTERM)
WRITE(6,510) (CVTER(I),I=1,NTERM)
WRITE(3,510) (CVTER(I),I=1,NTERM)
WRITE(6,511) (CCV(I),I=1,NTUBES)
WRITE(3,511) (CCV(I),I=1,NTUBES)
WRITE(6,520) CVTOT,CVTERM,RTOTSUM,SUM
WRITE(3,520) CVTOT,CVTERM,RTOTSUM,SUM
WRITE(6,530) (RTOT(I),I=1,NTERM)
WRITE(3,530) (RTOT(I),I=1,NTERM)
WRITE(6,540) (LP(I),I=1,NTUBES)
WRITE(3,540) (LP(I),I=1,NTUBES)
WRITE(6,550) (LQ(I),I=1,NTUBES)
WRITE(3,550) (LQ(I),I=1,NTUBES)
WRITE(6,570) TMAX/NTDIV,TMAX/NTDIV2,XRTOT,PV,PO,RHO,XMU,PM,
*DELP,PRESI
WRITE(3,570) TMAX/NTDIV,TMAX/NTDIV2,XRTOT,PV,PO,RHO,XMU,PM,
*DELP,PRESI
WRITE(6,580) NTDIV,NTDIV2,NTDIV3,NTDIV4,NFORCP,NPTSFFP,
*(PMULT(I),I=1,NFORCE)
C WRITE(6,590) (TLAG(I),I=1,NFORCE)
WRITE(6,600) HR,DX1,DX2,TMAX,NPERM,NPERL,NPERP,NPRIPP,TIMP
WRITE(6,653) (DX(I),I=1,NTUBES)
WRITE(3,580) NTDIV,NTDIV2,NTDIV3,NTDIV4,NFORCP,NPTSFFP,
*(PMULT(I),I=1,NFORCE)
C WRITE(3,590) (TLAG(I),I=1,NFORCE)
WRITE(3,600) HR,DX1,DX2,TMAX,NPERM,NPERL,NPERP,NPRIPP,TIMP
WRITE(3,653) (DX(I),I=1,NTUBES)
653 FORMAT(' DX=',20F6.3)
IF(ICONTIN.EQ.1) GO TO 6599
FAC=1.
WAVESP=254.7*(FAC*PO/ALFA/1333.)**.5
C WAVE SPEED FOR TUBE WITH DA/DP=B/P (RAINES)
CALCSP=DX1/DT
IF(DX1.LE.0.00001) CALCSP=DX(1)/DT
WRITE(6,650) WAVESP,CALCSP
WRITE(3,650) WAVESP,CALCSP
WRITE(6,651) ALFA,ISTEN,IANUR,JANUR,KANUR,DX1,DX2,
*NTDIV,NTDIV2,NPERM
WRITE(3,651) ALFA,ISTEN,IANUR,JANUR,KANUR,DX1,DX2,
*NTDIV,NTDIV2,NPERM
651 FORMAT(' **** AL=',F4.1,' ST-ANU=',4I1,' DX12=',
*2F6.3,' NDTIV=',2I6,' PERIOD=',I3)
I=NTANUR(1)
C NEEDS SOME GENERALIZING!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
IF(I.GE.1) WRITE(6,652) PALFA(1),I,LP(I),DX(I),D(I)
IF(I.GE.1) WRITE(3,652) PALFA(1),I,LP(I),DX(I),D(I)
652 FORMAT(' **** PAL=',F4.1,' VESSEL=',I4,' LENGHT=',
*I3,' *',F5.3,' DIAM=',F6.3)
IF(IRC.EQ.0) GO TO 6599

```

```

650 FORMAT(' ', 'NOMINAL WAVESPEED', F8.1, 5X, 'CALCULATION SPEED', F8.1)
      WRITE (6,660)
      WRITE (3,660)
660 FORMAT(1X, 'TUBE NO. RAD FREQ      C0      Z0      Z0      R1
1      R2/      ZT      ZT      REFLECTION COEFF. ')
      WRITE (6,670)
      WRITE (3,670)
670 FORMAT (1X, ' NO. HARM      REAL      IMAG      REAL      IMAG      MAG      PHASE')
1      R1      REAL      IMAG      REAL      IMAG      MAG      PHASE')
      DO 6500 J=1, NTUBES
      IF (TLINK(J).NE.0) THEN
      LLR=LP(J)
      RADIUS=(AO(J,LLR)/PI)**.5
      R1=XRTOT*RTOT(TLINK(J))
      R2=(1-XRTOT)*RTOT(TLINK(J))
      RFRAC=R2/R1
      CAPV=CVTER(TLINK(J))
      C0=AO(J,LLR)*ALFA/(ALN*PO)
C CALCULATE CHARACTERISTIC Z OF TUBE
      R0=8.*XMU/(PI*RADIUS**4)
      XL0=RHO/(PI*RADIUS**2)
      DO 6550 N=1, NOHRMS
      W=2.*PI*HR/60.*N
      ZOMAG=((XL0/C0)**2+(R0/(W*C0))**2)**.25
      THETA=-.5*ATAN(R0/(W*XL0))
      ZOR=ZOMAG*COS(THETA)
      ZOI=ZOMAG*SIN(THETA)
C CALCULATE THE Z OF R-C-R
      ZTR=R1+R2/(1.+R2**2*CAPV**2*W**2)
      ZTI=-R2*CAPV*W/(1.+R2**2*CAPV**2*W**2)
C CALCULATE REFLECTION COEFFICIENT
      XKRN=ZTR-ZOR
      XKIN=ZOI-ZTI
      XKRD=ZTR+ZOR
      XKID=ZTI+ZOI
      XRCRN=XKRN*XKRD-XKIN*XKID
      XRCIN=XKIN*XKRD+XKRN*XKID
      XRCD=XKRD**2+XKID**2
      XRCR=XRCRN/XRCD
      XRCI=-XRCIN/XRCD
      XRCMAG=SQRT(XRCR**2+XRCI**2)
      PHASE=ATAN(XRCIN/XRCRN)*57.0
      WRITE(6,690) J,N,RADIUS,W,C0,ZOR,ZOI
      WRITE(3,690) J,N,RADIUS,W,C0,ZOR,ZOI
1, R1, RFRAC, ZTR, ZTI
2, XRCR, XRCI, XRCMAG, PHASE
690 FORMAT(1X, 2I3, F5.3, 1X, F5.1, 1X, E9.3, 1X,
1E9.3, 1X, E9.3, 1X, E9.3, 1X,
21X, F5.2, 1X, E9.3, 1X,
3E9.3, 1X, E9.3, 1X, E9.3, 1X,
4F6.3, 1X, F6.2)
6550 CONTINUE
      ENDIF
6500 CONTINUE
C
C INITIALIZE COUNTERS
6599 KST=0

```



```

NPLOT=0
NVP=1
IT=0
C MECMEC=0
C IF (MECMEC.EQ.0) GO TO 99099
I1=0
IPLTC=0
NVP=0
ZT=0.0
C ANEURYSM CALCS
IF (IANUR.EQ.0) GO TO 6512
DO 96500 LL=1, IANUR
KK=NTANUR(LL)
C LINEAR(KK)=1
C#####ALL NONLINEAR CALC!!!!
PAN=1.+PANUR(LL)
DIA(KK,4)=PAN*DIA(KK,4)
DIA(KK,5)=DIA(KK,4)
DIA(KK,3)=.5*(DIA(KK,4)+DIA(KK,2))
DIA(KK,6)=DIA(KK,3)
AO(KK,4)=PI*DIA(KK,4)**2/4.
AO(KK,3)=PI*DIA(KK,3)**2/4.
AO(KK,5)=AO(KK,4)
AO(KK,6)=AO(KK,3)
PAA=1.+PALFA(LL)
AFA(KK,4)=PAA*AFA(KK,4)
AFA(KK,5)=AFA(KK,4)
AFA(KK,3)=0.5*(AFA(KK,4)+AFA(KK,2))
AFA(KK,6)=AFA(KK,3)
96500 CONTINUE
6512 CONTINUE
C DO FOR ALL PERIODS AND FOR ALL STEP TIMES DT IN INCREMENTS OF
C DT UP TO MAXIMUM TIME, TMAX.
CLOSE(4)
CLOSE(3)
IF (IBINARY.EQ.0) THEN
OPEN(UNIT=IP, FILE='WRP', STATUS='UNKNOWN', FORM='UNFORMATTED')
OPEN(UNIT=IQ, FILE='WRQ', STATUS='UNKNOWN', FORM='UNFORMATTED')
C OPEN(UNIT=IA, FILE='WRA', STATUS='UNKNOWN', FORM='UNFORMATTED')
ELSE
OPEN(UNIT=IP, FILE='WRP', STATUS='UNKNOWN', FORM='FORMATTED')
OPEN(UNIT=IQ, FILE='WRQ', STATUS='UNKNOWN', FORM='FORMATTED')
C OPEN(UNIT=IA, FILE='WRA', STATUS='UNKNOWN', FORM='FORMATTED')
ENDIF
CCTERM(NTERM+1)=CVTOT
IF (IBINARY.EQ.0) THEN
WRITE(7) NTUBES, ISTEN, IANUR, JANUR, KANUR, ALFA, ALFA1, PALFA(1)
*, PXMA, PXMI, PFCTOR, DFCTOR, RFCTOR, NST, NSTA, NSTB, NSTB1, KTSB
WRITE(7) NTSTEN, PSTEN, NODSTEN, NOSEGS
WRITE(7) NTANUR, PANUR, PALFA, XRTOT, NTERM, CCTERM
WRITE(8) NTUBES, ISTEN, IANUR, JANUR, KANUR, ALFA, ALFA1, PALFA(1)
*, PXMA, PXMI, PFCTOR, DFCTOR, RFCTOR, NTS, NTSB, NTSB1, KTSB
WRITE(8) NTSTEN, PSTEN, NODSTEN, NOSEGS
WRITE(8) NTANUR, PANUR, PALFA, XRTOT, NTERM, CCTERM
WRITE(IP) NTUBES, ISTEN, IANUR, JANUR, KANUR, ALFA, ALFA1, PALFA(1)
*, PXMA, PXMI, PFCTOR, DFCTOR, RFCTOR
WRITE(IP) (LP(I), I=1, NTUBES)

```

```

WRITE(IP) NTSTEN, PSTEN, NODSTEN, NOSEGS
WRITE(IP) NTANUR, PANUR, PALFA, XRTOT, NTERM, CCTERM
WRITE(IQ) NTUBES, ISTEN, IANUR, JANUR, KANUR, ALFA, ALFA1, PALFA(1)
*, PXMA, PXMI, PFCTOR, DFCTOR, RFCTOR
WRITE(IQ) (LQ(I), I=1, NTUBES)
WRITE(IQ) NTSTEN, PSTEN, NODSTEN, NOSEGS
WRITE(IQ) NTANUR, PANUR, PALFA, XRTOT, NTERM, CCTERM
ENDIF
39396 IF (IMM.GE.1) OPEN(UNIT=IMM, FILE='WRIMM', STATUS='UNKNOWN')
OPEN(UNIT=3, FILE='WRA', STATUS='UNKNOWN')
CALL SUBSCR(NTUBES, FLAG, LLINK, NTJ, ISIGN, LQJ, LPTB
*, LPJ, KJ, LQ, LP, RLINK, MLINK, TERMS, NTNTOT, PLINK, ISOURCE, KSOURC)
DO 37090 J=1, NTUBES
IF (TLINK(J).NE.0) THEN
ZC(J)=(1./AO(J,LP(J)))*SQRT(51.1*RHO*1333.*PZC/AFA(J,LP(J)))
RATIO=ZC(J)/RTOT(TLINK(J))
C ZC(J)=1.2*ZC(J)
WRITE(6,29319)
WRITE(3,29319)
29319 FORMAT(2X,'J,ZC(J),RTOT(TLINK(J)),RATIO FOLLOW')
WRITE(6,29219) J,ZC(J),RTOT(TLINK(J)),RATIO,AO(J,LP(J)),
1AFA(J,LP(J))
WRITE(3,29219) J,ZC(J),RTOT(TLINK(J)),RATIO,AO(J,LP(J)),
1AFA(J,LP(J))
ENDIF
37090 CONTINUE
29219 FORMAT (I5,5F15.5)
JJKK=0
C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! START OF 1ST BIG DO-LOOP
MM = 0
NNN = 0
MSKIP=0
DO 7000 NPER=1,NPERM
IF (NPER.EQ.1) CALL TIME(XYZ)
IF (NPER.EQ.1) WRITE(3,13132) NPER,X,YZ
13132 FORMAT('NPER & TIME XYZ=',I5,5X,A11)
IF (NPER.GT.NPERL) NTDIV=NTDIV2
IF (NPER.GT.NPERL+2) NTDIV=NTDIV3
IF (NPER.GT.NPERL+4) NTDIV=NTDIV4
DT=TMAX/NTDIV
PRINT *, 'NPER,NTDIV,DT', NPER,NTDIV,DT
C
91818 IF(MSKIP.EQ.2) MSKIP=0
DO 1801 J=1,NTUBES
PAVE(J)=0.
1801 QAVE(J)=0.
DO 78788 J=1,NTUBES
DO 78788 I=1,POINTS
PALL(J,I)=0.
78788 QALL(J,I)=0.
C
70017 M=0
KFJ=0
N=0
DO 65433 JAN=1,11
65433 PRINT *,MENOI
JANS=0

```

```

      PRINT *,MENO1,'          M.E.Clark'
      PRINT *,MENO1,'          *****'
      PRINT *,MENO1,'          *                      *'
DO 65438 JAN=1,5
65438 PRINT *,MENO1,'          * please, let it run ! *'
      PRINT *,MENO1,'          *                      *'
      PRINT *,MENO1,'          *****'
      WRITE(6,176) NTDIV2
   176 FORMAT(12X,'TIME DIVISION=',I6)
      PRINT *,MENO1
      DO 65434 JAN=1,3
65434 PRINT *,MENO1
      JAN=0
      PMAX=0.
      QMAX=0.
      AMAX=0.
      KCAMX1=NTDIV/KCAMX
      PMIN=500.
      QMIN=500.
      AMIN=500.
      KCA=0
      WRITE(6,16343)
16343 FORMAT(6X,'QMEAS1',4X,'QMEAS2',4X,'QMEAS3',4X,'QMEA4',
      14X,'QMEAS5',4X,'QMEAS6',4X,'QMEAS7')
      WRITE(6,56342) QMEAS(110),QMEAS(112),QMEAS(113),QMEAS(115),
      1QMEAS(117),QMEAS(119),QMEAS(121)
C      WRITE(6,56343)
c56343 FORMAT(6X,'QMOS24',4X,'QMOS(52)',4X,'QMOS25',4X,'QMOS53',
C      14X,'QMOS12',4X,'QMOS5',4X,'QMOS21')
      WRITE(6,56342) QAVG(110)*60.0,QAVG(112)*60.0,
      1QAVG(113)*60.0,QAVG(115)*60.0,
      1QAVG(117)*60.0,QAVG(119)*60.0,QAVG(121)*60.0
56342 FORMAT(2X,7F10.5)
      WRITE(6,65431) NPER,MEECCC,MMMM,MMCTRL
65431 FORMAT('                PERIOD:',I3,
      * 'MEECCC',I3,'MMMM=',I3,' MMCTRL',I3)
65432 FORMAT('+STEP',I6,F10.5)
      DO 7100 T=ZT,TMAX,DT
C
C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! START OF 2ND BIG DO-LOOP
C
      JAN=JAN+1
      JANS=JANS+1
      CYCTM=T/TMAX
      KFJ=KFJ+1
ccmec if(nper.ne.1) go to 40404
ccmec if(kfj.eq.ntdiv) go to 30303
C FOLLOWING IS SOME CODE TO CHECK ON FLOW CONTINUITY AT THE JUNCTIONS
      MECMEC=1
      IF(MECMEC.EQ.1) GO TO 40904
30303 read(19,30304) junct,jt1,jt2,jt3,jt4
      write(3,30304)junct,jt1,jt2,jt3,jt4,kfj
      write(6,30304)junct,jt1,jt2,jt3,jt4,kfj
      if(jt4.eq.1) go to 10101
      if(jt4.eq.2) go to 20202
      if(jt4.eq.3) go to 20202
30304 format(5i5)

```

```

10101 qjt1=q(jt1,lq(jt1))
      qjt2=q(jt2,1)
      qjt3=q(jt3,1)
      qjt4=qjt2+qjt3
      qdif=abs(qjt1-qjt4)
      write(3,30309)qjt1,qjt2,qjt3,qjt4,qdif
30309 format(2x,4f10.6,f15.12)
      write(6,30309)qjt1,qjt2,qjt3,qjt4,qdif
      if(qdif.gt.0.002) write(3,30308)junct
      if(qdif.gt.0.002) write(6,30308)junct
      if(qdif.gt.0.002) stop
      go to 30303
20202 qjt1=q(jt1,lq(jt1))
      qjt2=q(jt2,lq(jt2))
      qjt3=q(jt3,1)
      qjt4=qjt1+qjt2
      qdif=abs(qjt3-qjt4)
      write(3,30309)qjt1,qjt2,qjt3,qjt4,qdif
      write(6,30309)qjt1,qjt2,qjt3,qjt4,qdif
      if (qdif.gt.0.002) write(3,30308)junct
30308 format(2x,'error in junc ',i5)
      if (qdif.gt.0.002) write(6,30308) junct
      if (qdif.gt.0.002) stop
      if (jt4.eq.3) go to 40904
      go to 30303
c      print *, 't,jan,kfj',t,jan,kfj
40904 IF(ITAPER.NE.0.AND.NPER.EQ.1) THEN
      JJKK=JJKK+1
c      IF(JJKK.GE.NTDIV) GO TO 40404
      DO 57010 II=1,NTSPEC
      KK=NTSP(II)
      D(KK)=D(KK)*(NTDIV-JJKK)/NTDIV
      IF(D(KK).LE.DEND(KK)) D(KK)=DEND(KK)
      DTAPER(KK)=DTAPER(KK)*(NTDIV-JJKK)/NTDIV
      IF(DTAPER(KK).LE.DTPREND(KK)) DTAPER(KK)=DTPREND(KK)
      AO(KK,1)=(3.14159*D(KK)*D(KK))/4.
      DIA(KK,1)=SQRT(4.*AO(KK,1)/3.14159)
      L=LP(KK)
      DIF=AO(KK,1)-DTAPER(KK)*DTAPER(KK)*3.14159/4.
      DO 59829 J=2,L
      AO(KK,J)=AO(KK,1)-(J-1)*DIF/L
      DIA(KK,J)=SQRT(4.*AO(KK,J)/3.14159)
59829 CONTINUE
57010 CONTINUE
      if(jjkk.lt.100) write(6,110) (dia(ntspp(i),1),i=1,ntspec)
      if(jjkk.lt.100) write(3,110) (dia(ntspp(i),1),i=1,ntspec)
      ENDIF
40404 IF (ISTEN.NE.0.AND.NPER.EQ.1) THEN
c ESTABLISH STENOSIS SLOWLY OVER FIRST LINEAR PERIOD
      KST=KST+1
      IF (KST.GT.NTDIV) GO TO 99799
      DO 7010 II=1,ISTEN
c FOR TUBE WITH STENOSIS, LP MUST BE AT LEAST 8 (6?)
      KK=NTSTEN(II)
      IZ=NODSTEN(II)
      IF(KST.EQ.1)DIASAV(ii)=DIA(KK,IZ)
      IZM1=IZ-1

```

```

IZM2=IZ-2
IZP1=IZ+1
IZP2=IZ+2
IZP3=IZ+3
PST=1.-PSTEN(II)*KST/NTDIV
QST=1.-PSTEN(II)
IF(PST.LT.QST) PST=QST
DIA(KK,IZ)=PST*DIASAV(ii)
DIA(KK,IZM1)=0.5*(DIA(KK,IZM2)+DIA(KK,IZ))
DIA(KK,IZP1)=DIA(KK,IZ)
DIA(KK,IZP2)=0.5*(DIA(KK,IZP1)+DIA(KK,IZP3))
DO 70111 JJ=1,NOSEGS(II)
IF(IZP3.EQ.LQ(KK)) GO TO 70111
IZP1=IZP1+1
IZP2=IZP2+1
IZP3=IZP3+1
DIA(KK,IZP1)=DIA(KK,IZ)
DIA(KK,IZP2)=0.5*(DIA(KK,IZP1)+DIA(KK,IZP3))
70111 CONTINUE
DO 7011 JJ=IZM1,IZP2
7011 AO(KK,JJ)=(3.14156*DIA(KK,JJ)**2)/4.0
7010 CONTINUE
ENDIF
IF(ISTEN.NE.0.AND.NPER.EQ.1.AND.KST.EQ.NTDIV) THEN
DO 97010 II=1,ISTEN
KK=NTSTEN(II)
WRITE(6,87010) II,KK,LP(KK),LQ(KK)
WRITE(3,87010) II,KK,LP(KK),LQ(KK)
WRITE(6,77010) (DIA(KK,JXY),JXY=1,LP(KK))
WRITE(3,77010) (DIA(KK,JXY),JXY=1,LP(KK))
WRITE(6,77010) (AO(KK,JYX),JYX=1,LP(KK))
WRITE(3,77010) (AO(KK,JYX),JYX=1,LP(KK))
97010 CONTINUE
87010 FORMAT(2X,'STENOSIS DIA AND AREA FOLLOW',4I5)
77010 FORMAT(2X,8F10.5)
ENDIF
C CALCULATE THE TUBE CROSS SECTIONAL AREA
C AREA AND CAP ARE NOT A FUNCTION OF CURRENT PRESSURE
99799 IF (NPER.LE.NPERL) THEN
DO 7020 K=1,NTUBES
DO 7021 J=1,LQ(K)
A(K,J)=AO(K,J)
C AP0(K,J)=0.95*AO(K,J)
CANVAS
CANVAS
7020 CONTINUE
ELSE
C CAP=DA/DP, USED IN THE MASS BALANCE
C AREA AND CAP ARE A FUNCTION OF CURRENT PRESSURE
DO 7030 K=1,NTUBES
IF(LINEAR(K).EQ.1) THEN
DO 97021 J=1,LQ(K)
A(K,J)=AO(K,J)
CAP(K,J)=(AO(K,J)*AFA(K,J))/(PO*ALN)
97021 CONTINUE
ELSE
DO 7031 J=1,LQ(K)

```

```

C PPROP IS A CUTOFF VALUE ON RAINES PLOT TO KEEP PRESS AT A HIGH ENOUGH
C VALUE SO THAT LOG WILL NOT GO NEGATIVE
      PPROP=P(K,J)
      IF(PPROP.LT.PSTOP) PPROP=PSTOP
      ABCD=PPROP/PO
70004 A(K,J)=AO(K,J)*(1.+AFA(K,J)/ALN*LOG(ABCD))
      CAP(K,J)=AO(K,J)*AFA(K,J)/ALN/PPROP
      R(K,J)=8.*3.1416*XMU/A(K,J)**2
C      IF(LINEAR(K).EQ.1.AND.J.GE.IZM2.OR.J.LE.IZP2) A(K,J)=AO(K,J)
C      IF(LINEAR(K).EQ.1.AND.J.GE.IZM2.OR.J.LE.IZP2) CAP(K,J)=(AO(K,J)
C      1*AFA(K,J))/(PO*ALN)
7031  CONTINUE
      ENDIF
      7030 CONTINUE
      ENDIF
C      print *, '7030'
C
C CALCULATE THE PRESSURE FIRST FROM THE MASS BALANCE
C COMPUTE THE PRESSURES STARTING AT POINT NO.2 AND ENDING
C AT POINT LP-1 (I.E. POINTS INTERIOR TO THE TUBE). DOES
C NOT COMPUTE THE PRESSURES AT THE CENTER OF A JUNCTION AND
C THE PRESSURES PRODUCED BY PRESSURE SOURCES (I.E. DOES NOT
C COMPUTE THE PRESSURES AT THE ENDS OF THE TUBES)
C NOTE THAT FOR THE INLET TUBES FOR FLOW SOURCES THE INLET TUBES
C BEGIN AT POINT NO.1 AND END AT POINT LQ-1 (I.E., THE FIRST
C PRESSURE IS NOT GIVEN AND LQ=LP)
      IF(JAN.EQ.1000) WRITE(6,65432) JANS,T
      IF(JAN.EQ.1000) JAN=0
      DO 7040 K=1,NTUBES
        L=LQ(K)
        DO 7041 JJ=1,L
9041  PH(K,JJ)=P(K,JJ)
7040  CONTINUE
C      print *, '7040'
      IF(NPER.NE.NPERM) GO TO 13579
      IF(IWS.EQ.0) GO TO 93579
      IF(P(II1,JJ1).GT.PSAV22) TSAV22=T
13578 IF(P(II1,JJ1).GT.PSAV22) PSAV22=P(II1,JJ1)
      IF(P(II1,JJ1A).GT.PSAV28) TSAV28=T
13577 IF(P(II1,JJ1A).GT.PSAV28) PSAV28=P(II1,JJ1A)
      IF(P(II2,JJ2).GT.PSAV42) TSAV42=T
13576 IF(P(II2,JJ2).GT.PSAV42) PSAV42=P(II2,JJ2)
      IF(P(II2,JJ2A).GT.PSAV48) TSAV48=T
13575 IF(P(II2,JJ2A).GT.PSAV48) PSAV48=P(II2,JJ2A)
C      IF(NPER.EQ.NPERM) SAVP42(KFJ)=P(II2,JJ2)/1333.2
C      IF(NPER.EQ.NPERM) SAVP48(KFJ)=P(II2,JJ2A)/1333.2
93579 CONTINUE
13579 CONTINUE
C
C CALCULATE THE FLOWS AT A JUNCTION AND THE PRESSURE
C AT THE CENTER OF A JUNCTION. FLOWS AT A BIFURCATION
C SATISFY THE CONTINUITY EQUATION AND MOMENTUM BALANCE.
C IF TUBES COMES TOGETHER TO FORM A JUNCTION, FLAG=1
C SOLVE SIMULTANEOUSLY THE MASS BALANCE AT THE JUNCTION
C AND THE THREE MOMENTUM BALANCES AT THE TUBE ENDS CONNECTED
C TO THE TUBE JUNCTION FOR THE PRESSURE AT THE CENTER OF THE
C JUNCTION AND THE THREE FLOWS AT THE JUNCTION.

```

```

C CALCULATE JUNCTION PRESSURE
  DO 7050 NTN=1,NTNTOT
    NTJJ=NTJ(NTN)
    QARP=0.
    ARHOR=0.
C   IF (T.GT.0.0005.OR.T.LT.0.1) GO TO 19283
C   IF (J.NE.1.OR.J.NE.4.OR.J.NE.9) GO TO 19283
C   WRITE(6,100) J,NTJ,(LQJ(M),M=1,4),(LPTB(M),M=1,4),
C   1(LPJ(M),M=1,4),(ISIGN(M),M=1,4),(KJ(M),M=1,4)
19283 DO 7051 K=1,NTJJ
  KKJ=KJ(NTN,K)
  LLQJ=LQJ(NTN,K)
  LLPTB=LPTB(NTN,K)
  QARP=QARP+(ISIGN(NTN,K)*Q(KKJ,LLQJ)+A(KKJ,LLPTB)/
  *RHO*DT/DX(KKJ)*P(KKJ,LLPTB))/(1.+R(KKJ,LLPTB)*A(KKJ,
  *LLPTB)*DT/RHO)
  ARHOR=ARHOR+A(KKJ,LLPTB)/RHO*DT/DX(KKJ)/(1.+R(KKJ,
  *LLPTB)*A(KKJ,LLPTB)/RHO*DT)
C   IF(KFJ.GT.3.AND.NTN.GT.6) GO TO 7051
C   WRITE(6,98111) ARHOR,A(KKJ,LLPTB),DX(KKJ),R(KKJ,LLPTB),QARP,
C   1LLPTB,KKJ,NTJJ
C   WRITE(6,98119) NTN,LLQJ,ISIGN(NTN,K),Q(KKJ,LLQJ),P(KKJ,LLPTB)
C98111 FORMAT(5F10.5,3I10)
C98119 FORMAT(3I10,2F12.3)
  7051 CONTINUE
  PJ=QARP/ARHOR
C CONTINUITY OF PRESSURE AT THE CENTER OF THE JUNCTION
  DO 7052 K=1,NTJJ
    KKJ=KJ(NTN,K)
    LLPJ=LPJ(NTN,K)
    PH(KKJ,LLPJ)=PJ
    P(KKJ,LLPJ)=PJ
  7052 CONTINUE
C LINEAR MOMENTUM EQUATION AT THE JUNCTION
C NOTE-SPECIAL CASE FOR INLET TUBES OF FLOW SOURCES
C DUE TO FACT THAT LQ=LP. THEREFORE ADJUSTMENTS ARE
C NECESSARY TO MAINTAIN PROPER SUBSCRIPT SPECIFICATIONS
C NEW FLAG KSOURC(NTN) CREATED TO KEEP TRACK OF THOSE
C JUNCTIONS CONNECTED TO FLOW SOURCE VESSELS (SEE SUBR)
  DO 7053 K=1,NTJJ
    KKJ=KJ(NTN,K)
    LLQJ=LQJ(NTN,K)
    LLPTB=LPTB(NTN,K)
    Q(KKJ,LLQJ)=(Q(KKJ,LLQJ)-A(KKJ,LLPTB)/RHO*DT/DX(KKJ)*
    *(P(KKJ,LLQJ+1)-P(KKJ,LLQJ)))/(1.+
    *R(KKJ,LLPTB)*A(KKJ,LLPTB)*DT/RHO)
C   ENDIF
  7053 CONTINUE
  7050 CONTINUE
C*****
C INITIALIZE FORCING PRESSURES FOR INPUT TUBES USING SINE
C WAVE OR PHYSIOLOGICAL FORCING FUNCTION (AORTIC PULSE)
  IF(NFORCP.EQ.0) GO TO 17281
C   IF (NSINUO.NE.0) THEN
C   DO 7060 J=1,NTUBES
C   IF (PLINK(J).NE.0) THEN
C   P(J,1)=1333.*PAMP(PLINK(J))*SIN(6.2832*HR*T/60.)+PO

```

```

C      P(J,1)=1333.*PAMP(PLINK(J))*SIN(6.2832*HR*T/60.)+PO+PV
C      ENDIF
C 7060 CONTINUE
C      ELSE
C      MEC=0
C      DO 7070 J=1,NTUBES
C      IF (PLINK(J).NE.0) THEN
C      MEC=MEC+1
C      SCFG=(NPTSFFP-1)/(2.*TMAX)
C      K=(T+TMAX-TLAGP(MEC)*TMAX)*SCFG
C      I1=K+1
C      IF (I1-NPTSFFP) 1,2,2
2      I1=NPTSFFP-1
1      CONTINUE
C      I2=I1+1
C LINEAR INTERPOLATION
C      QP=(PFG(MEC,I1)+((PFG(MEC,I2)-PFG(MEC,I1))*((T+
* TMAX-TLAGP(MEC)*TMAX)*SCFG+1.-I1)))*PMULT(MEC)
C      P(J,1)=QP*1333.
C      P(J,1)=QP*1333.+PV
C      END IF
7070 CONTINUE
C      ENDIF
C INITIALIZE FORCING FLOWS FOR INPUT TUBES
C PHYSIOLOGICAL FORCING FUNCTION (MRI FLOW PULSE)
17281 IF(NFORCQ.EQ.0) GO TO 17282
C      MEC=0
C      DO 47070 J=1,NTUBES
C      IF (QLINK(J).NE.0) THEN
C      MEC=MEC+1
C      SCFG=(NPTSFFQ-1)/(2.*TMAX)
C      K=(T+TMAX-TLAGQ(MEC)*TMAX)*SCFG
C      I1=K+1
C      IF (I1-NPTSFFQ) 61,62,62
62      I1=NPTSFFQ-1
61      CONTINUE
C      I2=I1+1
C LINEAR INTERPOLATION
C      QP=(QFG(MEC,I1)+((QFG(MEC,I2)-QFG(MEC,I1))*((T+
* TMAX-TLAGQ(MEC)*TMAX)*SCFG+1.-I1)))*QMULT(MEC)
C      Q(J,1)=QP
C      ENDIF
47070 CONTINUE
17282 IF(NFORCT.EQ.0) GO TO 17283
C INITIALIZE TERMINAL FLOWS FOR SELECTED EFFERENT TUBES
C AS PHYSIOLOGICAL FORCING FUNCTIONS (MRI FLOW PULSE)
C      MEC=0
C      DO 57070 J=1,NTUBES
C      IF (TQLINK(J).NE.0) THEN
C      MEC=MEC+1
C      SCFG=(NPTSFFT-1)/(2.*TMAX)
C      K=(T+TMAX-TLAGT(MEC)*TMAX)*SCFG
C      I1=K+1
C      IF (I1-NPTSFFT) 71,72,72
72      I1=NPTSFFT-1
71      CONTINUE
C      I2=I1+1

```



```

C LINEAR INTERPOLATION
      QP= (TFG (MEC, I1) + ( (TFG (MEC, I2) -TFG (MEC, I1) ) * ( (T+
      * TMAX-TLAGT (MEC) *TMAX) *SCFG+1.-I1) ) ) *TMULT (MEC)
C
C INTRODUCING TERMINAL FLOW SOURCES
C
      QQPP (J)=QP
      ENDIF
57070 CONTINUE
17283 CONTINUE
C FLOW IS CALCULATED AT THE TERMINATIONS FROM THE MOMENTUM
C BALANCE LATER ON. AFTER THE PRESS HAS BEEN CALC BY CONTINUITY,
C ALL THE FLOWS ARE CALC FROM THE MOMENTUM BALANCE
C IF LP(K)=3, MOST LIKELY K IS A SHORT INTERNAL TUBE AND SO ALL FLOWS
C WILL BE CALC BY JUNC.EQ. BUT IT MIGHT BE A SHORT (LP.LE.3) TERMINAL
C IN WHICH CASE WE NEED TO CALC Q2 AND Q3 JUST BEFORE THE RCR
      DO 7080 K=1,NTUBES
      IF (LP(K).GT.3.OR.TLINK(K).NE.0) THEN
      L=LQ(K)-1
      JSTART=2
      IF (PLINK(K).NE.0) JSTART = 1
C CHECK TO SEE IF TUBE CONTAINS A STENOSIS. IF IT DOES, MAKE IT LINEAR
      ILINER =0
      IF (ISTEN.EQ.0) GO TO 77079
      DO 7079 IL=1,ISTEN
      IF (PSTEN(IL).EQ.0.) GO TO 7079
      IF (K.EQ.NTSTEN(IL)) ILINER=1
7079 CONTINUE
77079 CONTINUE
      IF (IANUR.EQ.0) GO TO 33079
      DO IL=1,IANUR
      IF (K.EQ.NTANUR(IL)) ILINER=1
      END DO
33079 CONTINUE
C IF (LINEAR(K).EQ.1) ILINER=1
C#####
      IF (NPER.LE.NPERL.OR.ILINER.EQ.1) THEN
C IF INLET TO TUBE IS A PRESSURE SOURCE, THE FIRST
C Q COMPUTED BY THE MOMENTUM BALANCE (FLOW) IS POINT
C NUMBER 1 AND JSTART=1. IF THE TUBE ENTRANCE CONTAINS A FLOW
C SOURCE OR IT IS ATTACHED TO A JUNCTION, THE FIRST
C Q COMPUTED BY THE MOMENTUM BALANCE (FLOW) IS AT
C POINT NUMBER 2 AND JSTART=2.
      DO 7081 J=JSTART,L
      tmp1 = DT/DX(K) * (A(K,J)+A(K,J+1)) / (2*RHO) * (P(K,J+1)-P(K,J))
      tmp2 = DT*A(K,J)*R(K,J)
      Q(K,J)=(Q(K,J)-tmp1)/(1.0+tmp2)
7081 CONTINUE
C VELOCITY PROFILE COMPUTED FOR ALL PERIODS GREATER THAN NPERL
      DO 7082 J=JSTART,L
C IF ANEURYSM, GO AROUND VEL PROF, BUT PICK UP CONV ACC
      IANURYES=0
      DO 56712 IL=1,IANUR
      IF (K.EQ.NTANUR(IL)) F(K,J)=R(K,J)
      IF (K.EQ.NTANUR(IL)) IANURYES=9
      IF (K.EQ.NTANUR(IL)) GO TO 97082
56712 CONTINUE

```

```

C COEFFICIENTS FOR A SIXTH DEGREE POLYNOMIAL APPROX.
C TO A VELOCITY PROFILE
  PGRAD=(PH(K,J+1)-PH(K,J))*A(K,J)/3.1416/(XMU*DX(K))
  VAVE=Q(K,J)/A(K,J)
  RSQD=A(K,J)/3.1416
  AV(K,J)=(-6.0*PGRAD+240.*VAVE+7.*RHO*RSQD/(XMU*DT)
  **AV(K,J))/(144.+7.*RHO*RSQD/(XMU*DT))
  AVP=AV(K,J)
  CVP=(PGRAD+240.*VAVE-144.*AVP)/(28.*RSQD)
  EVP=- (PGRAD+36.*AVP+32.*CVP*RSQD)/(20.*RSQD**2)
  GVP=- (AVP+CVP*RSQD+EVP*RSQD**2)/RSQD**3
  CV(K,J)=CVP
  EV(K,J)=EVP
  GV(K,J)=GVP
C CALCULATE WALL FRICTION FROM SLOPE OF VELOCITY
C PROFILE AT THE WALL
  F(K,J)=-XMU/RHO*A(K,J)*(4.*CVP+8.*EVP*RSQD+12.*GVP*RSQD**2)
97082 IF (J.EQ.1) THEN
C LINEAR MOMENTUM EQUATION FOR FIRST POINT FOR A PRESSURE SOURCE
  CONVAC=0.0
  ARGRAD=0.0
CANVAS
CANVAS
CANVAS
  ELSE
C UPWIND AND DOWNWIND DIFFERENCING OF CONVECTIVE ACC.
  IF (Q(K,J).GE.0.) CONVAC=Q(K,J)**2-Q(K,J-1)**2
  IF (Q(K,J).LT.0.) CONVAC=Q(K,J+1)**2-Q(K,J)**2
  IF (IANURYES.EQ.9) CONVAC=0.0
  ARGRAD=Q(K,J)/A(K,J)**2*(A(K,J+1)-A(K,J))
  IF (IANURYES.EQ.9) ARGRAD=0.0
  PGRAD=(A(K,J)+A(K,J+1))/(2*RHO)*(P(K,J+1)-P(K,J))
  Q(K,J)=(Q(K,J)-DT/DX(K)*(CONVAC/A(K,J)+PGRAD)
  * -DT*F(K,J))/(1.0-ARGRAD*DT/DX(K))
  ENDIF
7082 CONTINUE
  ENDIF
  ENDIF
7080 CONTINUE
C #####
  MMEC=0
  IF(MMEC.EQ.0) GO TO 23579
  IF(IWS.EQ.0) GO TO 23579
  IF(NPER.NE.NPERM) GO TO 23579
  IF(Q(II1,JJ1).GT.QSAV22) TSAT22=T
23578 IF(Q(II1,JJ1).GT.QSAV22) QSAV22=Q(II1,JJ1)
  IF(Q(II1,JJ1A).GT.QSAV28) TSAT28=T
23577 IF(Q(II1,JJ1A).GT.QSAV28) QSAV28=Q(II1,JJ1A)
  IF(Q(II2,JJ2).GT.QSAV42) TSAT42=T
23576 IF(Q(II2,JJ2).GT.QSAV42) QSAV42=Q(II2,JJ2)
  IF(Q(II2,JJ2A).GT.QSAV48) TSAT48=T
23575 IF(Q(II2,JJ2A).GT.QSAV48) QSAV48=Q(II2,JJ2A)
23579 CONTINUE
C
C CALCULATE THE FLOW AT THE END OF A TUBE WHICH IS TERMINATED
C BY A RESISTOR-BALLOON-RESISTOR FROM THE LAST PRESSURE POINT
C IN THE TUBE. XRTOT DIVIDES THE TOTAL RESISTANCE (RTOT)

```

```

C BETWEEN THE TWO RESISTORS.
C     IF(NPER.NE.NPERM) GO TO 54329
C     IF(T.EQ.DT) PRINT *,RTOT(1),RTOT(2),RTOT(3),RTOT(4),RTOT(5)
C     1,RTOT(6),RTOT(7),RTOT(8),RTOT(9),RTOT(10),RTOT(11),RTOT(12)
C     2,RTOT(13)
C TO IMPOSE CHARACTERISTIC IMPEDANCE AT ANY TERMINATION, USE FOLLOWING
C TWO STMTS AND RECALC Q AFTER STMT 7090 OR CALC NEW RTOT AS IN DO-LOOP
C 1000 AND LET DO-LOOP 7090 CALC Q
C     ZC=(1./AO(1,7))*SQRT(51.1*RHO*1333.*PZC/AFA(1,7))
C     Q(1,7)=(P(1,7)-1333.*PZC)/ZC
54329 DO 7090 J=1,NTUBES
      IF (IANUR.EQ.0) GO TO 97090
      DO 97091 IL=1,IANUR
        IF(J.EQ.NTANUR(IL)) Q(J,LQ(J))=0.0
97091 CONTINUE
97090 IF (TLINK(J).NE.0) THEN
C IF IZC=0 USE RCR W/XRTOT, IF =1 USE RCR W/ R1=ZC
      RZ=RTOT(TLINK(J))-ZC(J)
      IF(IZC.EQ.0) THEN
CANVAS
        *      *RTOT(TLINK(J))/((1.-XRTOT)*RTOT(TLINK(J)))
        *      +2.*CVTER(TLINK(J))/DT*(P(J,LQ(J))-PH(J,LQ(J)))
        *      +XRTOT*RTOT(TLINK(J))*Q(J,LQ(J)))/(1.+(XRTOT/
        *      (1.-XRTOT))+2.*CVTER(TLINK(J))*XRTOT*
        *      RTOT(TLINK(J))/DT)
        ELSE
        IF(RZ.GT.0.) THEN
CANVAS
        *      *RTOT(TLINK(J))/(RZ)
        *      +2.*CVTER(TLINK(J))/DT*(P(J,LQ(J))-PH(J,LQ(J)))
        *      +ZC(J)*Q(J,LQ(J)))/(1.+ZC(J)/RZ)
        *      +2.*CVTER(TLINK(J))*ZC(J)/DT)
        ELSE
CANVAS
        *      *RTOT(TLINK(J))/((1.-XRTOT)*RTOT(TLINK(J)))
        *      +2.*CVTER(TLINK(J))/DT*(P(J,LQ(J))-PH(J,LQ(J)))
CANVAS
        *      (1.-XRTOT))+2.*CVTER(TLINK(J))*XRTOT*RTOT(TLINK(J))/DT)
        ENDIF
        ENDIF
        ELSE IF(TQLINK(J).NE.0) THEN
        Q(J,LQ(J))=QQPP(J)
        ENDIF
7090 CONTINUE
      DO 6017 J=1,NTUBES
        PAVE(J)=PAVE(J)+P(J,2)
6017 QAVE(J)=QAVE(J)+Q(J,2)
      DO 78789 J=1,NTUBES
        DO 78771 I=1,LP(J)
78771 PALL(J,I)=PALL(J,I)+P(J,I)
        DO 78772 I=1,LQ(J)
78772 QALL(J,I)=QALL(J,I)+Q(J,I)
78789 CONTINUE
C
C WRITE PRESSURES AND FLOWS TO OUTPUT FILES WHEN ON PRINT
C CYCLE(T=MULTIPLE OF TP) FOR LAST NPER-NPERP CYCLES
      IF (NPER.GE.NPERP) THEN

```

```

      IF (T.GE.(M*TP)) THEN
      M=M+1
      CYCTM=T/TMAX
C      IF (M.LT.6) WRITE(6,170) CYCTM
      IF (IBINARY.EQ.0) THEN
      WRITE(IP)      CYCTM
      WRITE(IQ)      CYCTM
C      WRITE(IA)      CYCTM
      ELSE
      WRITE(IP,170) CYCTM
      WRITE(IQ,170) CYCTM
C      WRITE(IA,170) CYCTM
      ENDIF
13571 DO 7095 J=1,NTUBES
      DO 7096 K=1,LP(J)
      7096 PMM(J,K)=P(J,K)/1333.2
      IF (IFRIC.NE.0.AND.NPER.EQ.NPERM) THEN
C      WRITE(IIM,171) CYCTM
C      WRITE(IIM,180)
      DO 7097 K=1,LQ(J)
      RADIUS=SQRT(A(J,K)/3.1416)
      SHEARP=R(J,K)*Q(J,K)/(6.2832*RADIUS)
      SHEARN=-F(J,K)*RHO/(6.2832*RADIUS)
C      WRITE(IIM,190) J,SHEARN,SHEARP
      7097 CONTINUE
      ENDIF
      IF (IBINARY.EQ.0) THEN
      DO 44441 K=1,LP(J)
      44441 BB(K)=PMM(J,K)
      WRITE(IP)(BB(K),K=1,LP(J))
      DO 44442 K=1,LQ(J)
      44442 BB(K)=Q(J,K)
      WRITE(IQ)(BB(K),K=1,LQ(J))
C      WRITE(IA)(A(J,K),K=1,LP(J))
      ELSE
      WRITE(IP,200)(PMM(J,K),K=1,LP(J))
      WRITE(IQ,99200)(Q(J,K),K=1,LQ(J))
      ENDIF
      7095 CONTINUE
C      WRITE(6,276) CYCTM,(PMM(1,K),K=1,LP(1)),(PMM(5,K),K=1,LP(5))
      276 FORMAT(2X,F8.6,13F8.3)
      IF (NPROFL.NE.0) THEN
      DO 7098 IN=1,NPROFL
      NT=NTPROF(IN)
      L=LPROF(IN)
      ETA=0.0
      DO 7099 NPT=1,30
      VP(NPT)=1.+CV(NT,L)/AV(NT,L)*ETA**2+EV(NT,L)/
      *AV(NT,L)*ETA**4+GV(NT,L)/AV(NT,L)*ETA**6
C      ETA=INCREMENTAL CHANGE IN R
      ETA=ETA+.035
      7099 CONTINUE
      WRITE(IIM,210) NT,L
      WRITE(IIM,220) (VP(NPT),NPT=1,30)
      NPLOT=NPLTC+1
      IF (NPLOT.EQ.5) THEN
      IPLTC=0

```

```

DO 7105 J=1,30
IPLTC=IPLTC+1
ETA=0.0
VCLPLT(IN,IPLTC,NVP)=AV(NT,L)+CV(NT,L)*ETA**2
*+EV(NT,L)*ETA**4+GV(NT,L)*ETA**6
ETA=ETA+.035
NVP=NVP+1
7105 CONTINUE
ENDIF
7098 CONTINUE
ENDIF
ENDIF
IF (NPER.EQ.NPERM.AND.T.GE.(N*TI)) THEN
N=N+1
CCYCTM(N)=T/TMAX
C SAVE THE PRESS AND FLOWS AT THE TERMINATIONS OVER THE ENTIRE
C PERIOD TO CALCULATE THE IMPEDENCE AT THE TERMINATIONS
IF (NPSAVE.NE.0) THEN
DO 7106 J=1,NPSAVE
K=NPTSVE(J)
JN=MPTSVE(J)
PP(J,N)=P(K,JN)/1333.
QQ(J,N)=Q(K,JN)
7106 CONTINUE
ITMAX=N
ENDIF
ENDIF
C
170 FORMAT('0','PERCENT OF CYCLE=',F8.6)
171 FORMAT('0','PERCENT OF CYCLE=',F8.6,2X,'ALFA=',
*3F5.1,2X,'3P+D+R=',2F4.0,3F5.2)
172 FORMAT(' ALFA=',3F5.1,2X,'3P+D+R=',2F4.0,3F5.2)
180 FORMAT(' ','TUBE NO.',5X,'WALL SHEAR, 6DEGREE',
*5X,'WALL SHEAR, 2DEGREE')
190 FORMAT(' ',I10,10X,E12.5,10X,E12.5)
200 FORMAT(13F8.3)
98710 FORMAT(13F8.5)
99200 FORMAT (13F9.4)
99300 FORMAT(13F9.5)
210 FORMAT(' ','TUBE NO.',I4,5X,'POINT NO.',I5,
*5X,'VELOCITY PROFILE')
220 FORMAT(' ',11F10.4)
ENDIF
LL=LP(ITEST)
LLL=LQ(ITEST)
IF (ITEST.GT.0.AND.NPER.GT.NPERL) THEN
WRITE(3,99201) KFJ,(PMM(ITEST,J),J=1,LL)
WRITE(3,99200) (Q(ITEST,J),J=1,LLL)
ENDIF
99201 FORMAT(I5,(13F7.2))
C FOLLOWING PRINT OF ANUR NEEDS TO BE REMOVED OR GENERALIZED
C IF(IANUR.EQ.0) GO TO 7100
C DO 97500 KK=1,IANUR
C IF(K.NE.NTANUR(1)) GO TO 7505
C IF(K.NE.NTANUR(KK)) GO TO 7505
C LL=LP(NST)
C IF(NPER.NE.NPERM) GO TO 7100

```

```

IF(NST.LE.0) GOTO 7501
PPMM=P(NST,NSTA+2)/1333.2
IF(PMAX.LT.PPMM) PMAX=PPMM
IF(QMAX.LT.Q(NST,NSTA+2)) QMAX=Q(NST,NSTA+2)
AA(1)=SQRT(A(NST,NSTA+2)*4./PI)
IF(AMAX.LT.AA(1)) AMAX=AA(1)
IF(PMIN.GT.PPMM) PMIN=PPMM
IF(QMIN.GT.Q(NST,NSTA+2)) QMIN=Q(NST,NSTA+2)
IF(AMIN.GT.AA(1)) AMIN=AA(1)
DO 7504 L=NSTA,NSTB
PMM(NST,L)=P(NST,L)/1333.
7504 CONTINUE
DO 7604 L=NTSA,NTSB
PMM(NTS,L)=P(NTS,L)/1333.
7604 CONTINUE
7501 KCA=KCA+1
IF(KCA.NE.KCAMX1) GO TO 7100
IF(NST.LE.0) GOTO 7506
DO 7502 L=NSTA,NSTB
PMM(NST,L)=P(NST,L)/1333.
7502 CONTINUE
7506 IF(NST.GT.0) THEN
IF(IBINARY.EQ.0) THEN
DO 44443 J=NSTA,NSTB
44443 BB(J)=PMM(NST,J)
WRITE(7) (BB(J),J=NSTA,NSTB)
DO 44444 J=NSTA,NSTB1
44444 BB(J)=Q(NST,J)
WRITE(7) (BB(J),J=NSTA,NSTB1)
DO 99210 J=NSTA,NSTB1
99210 BB(J)=SQRT(A(NST,J)*4./PI)
WRITE(7) (BB(J),J=NSTA,NSTB1)
ELSE
WRITE(7,200) (PMM(NST,J),J=NSTA,NSTB)
WRITE(7,99200) (Q(NST,J),J=NSTA,NSTB1)
WRITE(7,99300) (A(NST,J),J=NSTA,NSTB1)
ENDIF
ENDIF
IF(NTS.LE.0) GOTO 7606
DO 7602 L=NTSA,NTSB
PMM(NTS,L)=P(NTS,L)/1333.
7602 CONTINUE
7606 IF(NTS.GT.0) THEN
IF(IBINARY.EQ.0) THEN
DO 44445 J=NTSA,NTSB1,KTSB
44445 BB(J)=PMM(NTS,J)
WRITE(8) (BB(J),J=NTSA,NTSB,KTSB)
DO 44446 J=NTSA,NTSB1,KTSB
44446 BB(J)=Q(NTS,J)
WRITE(8) (BB(J),J=NTSA,NTSB1,KTSB)
DO 99211 J=NTSA,NTSB1,KTSB
99211 BB(J)=SQRT(A(NTS,J)*4./PI)
WRITE(8) (BB(J),J=NTSA,NTSB1,KTSB)
ELSE
WRITE(8,200) (PMM(NTS,J),J=NTSA,NTSB,KTSB)
WRITE(8,99200) (Q(NTS,J),J=NTSA,NTSB1,KTSB)
WRITE(8,99300) (A(NTS,J),J=NTSA,NTSB1,KTSB)

```

```

ENDIF
ENDIF
KCA=0
7100 CONTINUE
DO 1800 J=1,NTUBES
  QAVE(J)=QAVE(J)/NTDIV
  QAVG(J) = QAVE(J)
1800 PAVE(J)=PAVE(J)/1333.2/NTDIV
DO 78775 J=1,NTUBES
DO 78773 I=1,LP(J)
78773 PALL(J,I) = PALL(J,I)/1333.2/NTDIV
DO 78774 I=1,LQ(J)
78774 QALL(J,I) = QALL(J,I)/NTDIV
78775 CONTINUE
WRITE(3,370)
WRITE(3,380) (PAVE(J),J=1,NTUBES)
WRITE(3,390)
WRITE(3,380) (QAVE(J)*60.,J=1,NTUBES)
IF(MMCTRL.EQ.1.AND.NPER.EQ.NPERM) THEN
GO TO 91817
END IF
IF(MMCTRL.GT.1.AND.NPER.EQ.NPERM) THEN
CONTINUE
END IF
WRITE(6,16343)
WRITE(6,56342) QMEAS(110),QMEAS(112),QMEAS(113),QMEAS(115),
*QMEAS(117),QMEAS(119),QMEAS(121)
WRITE(6,56342) QAVG(110)*60.0,QAVG(112)*60.0,
*QAVG(113)*60.0,QAVG(115)*60.0,
*QAVG(117)*60.0,QAVG(119)*60.0,QAVG(121)*60.0
C
WRITE(IMEIDE,16343)
WRITE(IMEIDE,56342) QMEAS(110),QMEAS(112),QMEAS(113),QMEAS(115),
*QMEAS(117),QMEAS(119),QMEAS(121)
WRITE(IMEIDE,56342) QAVG(110)*60.0,QAVG(112)*60.0,
*QAVG(113)*60.0,QAVG(115)*60.0,
*QAVG(117)*60.0,QAVG(119)*60.0,QAVG(121)*60.0
C
C UPDATE RCR'S TO MEET MR FLOWS
C
C IF(NPER.LT.4) WRITE(IMEIDE,22918)NPER, NPERM,MMCTRL
22918 FORMAT(2X,'NPER,NPERM,MMCTRL-NO LENGTH ADJUST',3I5)
C IF(NPER.LT.4) GOTO 91889
C IF(NPER.GE.4) WRITE(IMEIDE,22917)NPER,NPERM,MMCTRL
22917 FORMAT(2X,'NPER,NPERM,MMCTRL-ADJUST LENGTH',3I5)
DO 99999 J=1,NTUBES
IF(ZLINK(J).NE.0) THEN
CANVAS
CANVAS
WRITE(IMEIDE,51629) J,ZLINK(J),QMOQA(ZLINK(J)),QMEAS(J),
*QAVE(J)*60.
51629 FORMAT(2X,'J,ZLINK,REAL QMOQA QM QA*60',2I5,3F10.3)
19225 FORMAT(2X,'J,TQL,QMOQA',2I5,F10.5)
END IF
99999 CONTINUE
MMMM = 0
DO I=1,NTUBES

```

```

      LLL=ZLINK(I)
      KKK=TLINK(I)
      IF (LLL.NE.0) THEN
        N=I
        XLQTOLD=XLTERM(KKK)
C      IF (QMOQA (LLL) .LT.0.75) QMOQA (LLL)=0.75
C      IF (QMOQA (LLL) .GT.1.25) QMOQA (LLL)=1.25
        WRITE (IMEIDE,33834) MMCTRL,N,QMEAS(I),QAVE(I)*60.,QMOQA (LLL)
33834 FORMAT(2X,'MMCTRL,N,QMEAS,QAVE,QMOQA',2I5,3F10.3)
CANVAS
CANVAS
      WRITE (IMEIDE,33804) N,XLQTOLD,XLTERM(KKK),QMOQA (LLL)
33804 FORMAT(2X,'OLD,NEW XLTERM AND RATIO',I5,3F10.3)
      ELSE
        MMMM = MMMM + 1
        WRITE (IMEIDE,12567) N
12567 FORMAT(2X,I5,' SECTOR OK')
      END IF
      END IF
      END DO
      WRITE (IMEIDE,7771) (XLTERM(I),I=1,17)
      WRITE (IMEIDE,7772) (DMTERM(I),I=1,17)
      WRITE (IMEIDE,7773) (QSTEDY(I),I=1,17)
C      IF (MMMM.EQ.7) GOTO 91889
      DO N=1,NTUBES
        MMM=ZLINK(N)
        IF (MMM.NE.0) THEN
          RTOT(MMM)=128.*XMU*XLTERM(MMM)/(3.1416*
*DMTERM(MMM)**4) - RTUBE(MMM)
C      WRITE (IMEIDE,1625) MMCTRL,N,RTOT(MMM),XLTERM(MMM),
C      *QMEAS(N),QAVE(N)*60.
1625 FORMAT(2X,'MMCTRL,N,RQT,XLQT,QM,QA',2I5,4F10.2)
C      WRITE (IMEIDE,66566) N,RTOT(MMM),RTUBE(MMM)
      END IF
      END DO
      WRITE (IMEIDE,44117) MMCTRL
44117 FORMAT(2X,'JUST ADJUSTED RTOT, MMCTRL',I5)
91889 IF (MMMM.EQ.7) THEN
51815 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3,5X,'LMCMEA',F10.4,2X,
1'LMCCALC',F10.4)
51816 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3,5X,'RMCMEA',F10.4,2X,
1'RMCCALC',F10.4)
51817 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3,5X,'LECMEA',F10.4,2X,
1'LECCALC',F10.4)
51818 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3,5X,'RECMEA',F10.4,2X,
1'RECCALC',F10.4)
51819 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3,5X,'ACMEA',F10.4,2X,
1'ACCALC',F10.4)
51820 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3,5X,'BAMEA',F10.4,2X,
1'BACALC',F10.4)
51822 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3,5X,'LVAMEA',F10.4,2X,
1'LVACALC',F10.4)
51823 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3,5X,'RVAMEA',F10.4,2X,
1'RVACALC',F10.4)
51824 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3,5X,'LCCMEA',F10.4,2X,
1'LCCALC',F10.4)
51825 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3,5X,'RCCMEA',F10.4,2X,

```



```

1'RCCALC',F10.4)
51826 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3,5X,'LICNMEA',F10.4,2X,
1'LICNCALC',F10.4)
51827 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3,5X,'RICNMEA',F10.4,2X,
1'RICNCALC',F10.4)
51828 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3,5X,'LICIMEA',F10.4,2X,
1'LICICALC',F10.4)
51829 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3,5X,'RICIMEA',F10.4,2X,
1'RICICALC',F10.4)
51821 FORMAT(2X,I5,2F10.4,I5,2F10.4,1X,2F6.3)
DO J=1,NTUBES,2
  IF(J.EQ.1)WRITE(20,51815) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5,QMEAS
2(110),QAVE(110)*60.
  IF(J.EQ.3)WRITE(20,51816) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5,QMEAS
2(112),QAVE(112)*60.
  IF(J.EQ.5)WRITE(20,51817) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5,QMEAS
2(115),QAVE(115)*60.
  IF(J.EQ.7)WRITE(20,51818) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5,QMEAS
2(117),QAVE(117)*60.
  IF(J.EQ.9)WRITE(20,51819) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5,QMEAS
2(113),QAVE(113)*60.
  IF(J.EQ.11)WRITE(20,51820) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5,QMEAS
2(5),QAVE(5)*60.
  IF(J.EQ.13)WRITE(20,51822) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5,QMEAS
2(2),QAVE(2)*60.
  IF(J.EQ.15)WRITE(20,51823) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5,QMEAS
2(1),QAVE(1)*60.
  IF(J.EQ.17)WRITE(20,51824) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5,QMEAS
2(14),QAVE(14)*60.
  IF(J.EQ.19)WRITE(20,51825) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5,QMEAS
2(13),QAVE(13)*60.
  IF(J.EQ.21)WRITE(20,51826) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5,QMEAS
2(16),QAVE(16)*60.
  IF(J.EQ.23)WRITE(20,51827) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5,QMEAS
2(15),QAVE(15)*60.
  IF(J.EQ.25)WRITE(20,51828) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5,QMEAS
2(18),QAVE(18)*60.
  IF(J.EQ.27)WRITE(20,51829) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5,QMEAS
2(17),QAVE(17)*60.
  IF(J.GT.27)WRITE(20,51821) J,QAVE(J)*60,PAVE(J),J+1,QAVE(J+1)*60,
1PAVE(J+1),(D(J)+DTAPER(J))*0.5,(D(J+1)+DTAPER(J+1))*0.5
END DO
MEECCC=MEECCC+1

```

```

WRITE(6,33881) MEECCC
33881 FORMAT(2X,'MMMM=7, EITHER ADJUST AND GO BACK OR QUIT!! MEECCC=',
1I5)
WRITE(IMEIDE,33881)MEECCC
IF(MEECCC.LT.6) THEN
IF(QAVE(11).LT.0.0.AND.QAVE(12).LT.0.0) THEN
QMEAS(121)=QMEAS(5)+ABS(QAVE(11)*60.)+ABS(QAVE(12)*60.)
END IF
IF(QAVE(11).GE.0.0.AND.QAVE(12).GE.0.0) THEN
QMEAS(121)=QMEAS(5)-ABS(QAVE(11)*60.)-ABS(QAVE(12)*60.)
END IF
IF(QAVE(11).LT.0.0.AND.QAVE(12).GE.0.0) THEN
QMEAS(121)=QMEAS(5)+ABS(QAVE(11)*60.)-ABS(QAVE(12)*60.)
END IF
IF(QAVE(11).GE.0.0.AND.QAVE(12).LT.0.0) THEN
QMEAS(121)=QMEAS(5)-ABS(QAVE(11)*60.)+ABS(QAVE(12)*60.)
END IF
IF(QAVE(36).LT.0.0) THEN
QMEAS(115)=QMEAS(24)-ABS(QAVE(36)*60.)
END IF
IF(QAVE(36).GE.0.0) THEN
QMEAS(115)=QMEAS(24)+ABS(QAVE(36)*60.)
END IF
IF(QAVE(37).LT.0.0) THEN
QMEAS(117)=QMEAS(25)-ABS(QAVE(37)*60.)
END IF
IF(QAVE(37).GE.0.0) THEN
QMEAS(117)=QMEAS(25)+ABS(QAVE(37)*60.)
END IF
WRITE(IMEIDE,13161) QMEAS(121),QMEAS(115),QMEAS(117)
13161 FORMAT(2X,'NEW VALUES OF QMEAS(121),(115),(117)',3F10.3)
MMMM=0
GO TO 91817
ELSE
WRITE(IMEIDE,7771) (XLTERM(I),I=1,17)
WRITE(IMEIDE,7772) (DMTERM(I),I=1,17)
WRITE(IMEIDE,7773) (QSTEDY(I),I=1,17)
WRITE(3,7771) (XLTERM(I),I=1,17)
WRITE(3,7772) (DMTERM(I),I=1,17)
WRITE(3,7773) (QSTEDY(I),I=1,17)
WRITE(6,7771) (XLTERM(I),I=1,17)
WRITE(6,7772) (DMTERM(I),I=1,17)
WRITE(6,7773) (QSTEDY(I),I=1,17)
STOP
END IF
END IF
C WRITE(IMEIDE,44115) MMMM,MMCTRL,NPERM
C44115 FORMAT(2X,'MMMM,MMCTRL,NPERM',3I5)
7771 FORMAT(8F10.1)
7772 FORMAT(8F10.1)
7773 FORMAT(8F10.1)
IF(NPER.EQ.NPERM.AND.MMCTRL.LE.10)GO TO 91817
81817 WRITE(3,391) NST
391 FORMAT(2X,'PRESSURE, FLOW & DIAMETER IN NST=',I3)
IF(NST.GT.0) WRITE(3,200) (P(NST,J)/1333.2,J=NSTA,NSTB)
IF(NST.GT.0) WRITE(3,99200) (Q(NST,J),J=NSTA,NSTB)
IF(NST.GT.0)WRITE(3,99300) (SQRT(A(NST,J)*4./PI),J=NSTA,NSTB-1)

```

```

      IF(NST.GT.0)WRITE(3,392) PMIN,PMAX,QMIN,QMAX,AMIN,AMAX
392  FORMAT(' ****  MIN-MAX VALUES:',2F8.3,2(2X,2F8.4))
C
C  INITIALIZE CENTERLINE VELOCITY TO COMPUTE
C  THE VELOCITY PROFILE
      IF (NPER.EQ.NPERL) THEN
        DO 7500 K=1,NTUBES
          L=LQ(K)-1
          LLL=LQ(K)
          LL=LP(K)
          JSTART=2
          IF (L.EQ.1.OR.PLINK(K).NE.0) JSTART=1
          DO 7510 I=JSTART,L
            AV(K,I)=1.5*Q(K,I)/A(K,I)
7510  CONTINUE
7500  CONTINUE
        ENDIF
C      WRITE(6,230) NPER
      IF (NPER.GE.NPERP) THEN
C      NLines(NPER-NPERM+2)=NPRIPP
      WRITE(3,230) NPER,NPRIPP
      ELSE
      WRITE(3,230) NPER,0
      ENDIF
230  FORMAT('0',5X,'PER NO.=' ,I2,2X,'PLTS/PRTS PER PERIOD=' ,I4)
240  FORMAT(' ', 'NO. OF PLOTS PER PERIOD',I4)
      CALL TIME(XYZ)
      WRITE(3,13132) NPER,XYZ
7000  CONTINUE
      IF(IWS.EQ.0) GO TO 99739
      PSAV22=PSAV22/1333.2
      PSAV28=PSAV28/1333.2
      PSAV42=PSAV42/1333.2
      PSAV48=PSAV48/1333.2
C      PRINT 737,PSAV22,TSAB22,PSAV28,TSAB28
C      PRINT 737,PSAV42,TSAB42,PSAV48,TSAB48
C      PRINT 738,QSAV22,TSAT22,QSAV28,TSAT28
C      PRINT 738,QSAV42,TSAT42,QSAV48,TSAT48
      IF(MMEC.EQ.0) GO TO 99739
      L1P=(JJ1A-JJ1)*DX(II1)
      L2P=(JJ2A-JJ2)*DX(II2)
      T1P=TSAB28-TSAB22
      T2P=TSAB48-TSAB42
      T1Q=TSAT28-TSAT22
      T2Q=TSAT48-TSAT42
      WS1P=L1P/T1P
      WS2P=L2P/T2P
      WS1Q=L1P/T1Q
      WS2Q=L2P/T2Q
C      PRINT 739,WS1P,WS2P,WS1Q,WS2Q
737  FORMAT (2X,2(F10.3,F10.6))
736  FORMAT (2X,15F6.1)
C      WRITE(6,736) (SAVP42(IJK),IJK=1,KFJ)
C      WRITE(6,736) (SAVP48(IJK),IJK=1,KFJ)
738  FORMAT (2X,2(F10.6,F10.6))
739  FORMAT (2X,'WAVE SPEEDS (P1,P2,Q1,Q2) FOLLOW',4F12.4)
C

```

```

99739 IF (NPSAVE.NE.0) THEN
    DO 8000 J=1,NPSAVE
    DO 8100 M=1,ITMAX
    PPLOT(M)=PP(J,M)
    PQ(M)=QQ(J,M)
    PR(M)=PP(J,M)
8100 CONTINUE
C FOURIER ANALYSIS OF PRESS AND FLOW AT THE TERMINATIONS
C IN ORDER TO CALCULATE THE INPUT IMPEDENCE AT THE TERMS.
    IF(IFORIER.EQ.0) GO TO 8000
    ODD=ITMAX/2.0
    IEVEN=ITMAX/2
    IF (ODD.NE.IEVEN) ITMAX=ITMAX-1
    ITMHAF=ITMAX/2/5
    DO 8200 N=1,ITMHAF
    SUMFCC=0.0
    SUMPCC=0.0
    SUMFCS=0.0
    SUMPCS=0.0
    DO 8210 I=1,ITMAX
    X=3.1416/IEVEN*I*N*1.00000000
    SUMFCS=SUMFCS+PQ(I)*SIN(X)
    SUMPCS=SUMPCS+PR(I)*SIN(X)
    SUMFCC=SUMFCC+PQ(I)*COS(X)
    SUMPCC=SUMPCC+PR(I)*COS(X)
8210 CONTINUE
    FCQS(N)=SUMFCS/IEVEN
    FCQC(N)=SUMFCC/IEVEN
    FCPS(N)=SUMPCS/IEVEN
    FCPC(N)=SUMPCC/IEVEN
    PHZP(N)=ATAN(FCPC(N)/FCPS(N))*57.3
    PHZQ(N)=ATAN(FCQC(N)/FCQS(N))*57.3
8200 CONTINUE
    DO 8300 M=1,ITMHAF
8300 ZF(M)=SQRT((FCPC(M)**2+FCPS(M)**2)/(FCQC(M)**2+FCQS(M)**2))
    WRITE(IIM,250) ITMHAF
    WRITE(IIM,260) (ZF(M),M=1,ITMHAF)
    WRITE(IIM,270)
    WRITE(IIM,280) (FCPC(I),I=1,ITMHAF)
    WRITE(IIM,290)
    WRITE(IIM,280) (FCPS(I),I=1,ITMHAF)
    WRITE(IIM,300)
    WRITE(IIM,310) (PHZP(I),I=1,ITMHAF)
    WRITE(IIM,9270)
    WRITE(IIM,280) (FCQC(I),I=1,ITMHAF)
    WRITE(IIM,9290)
    WRITE(IIM,280) (FCQS(I),I=1,ITMHAF)
    WRITE(IIM,9300)
    WRITE(IIM,310) (PHZQ(I),I=1,ITMHAF)
    WRITE(IIM,320)
    WRITE(IIM,330) (PP(J,IT),IT=1,ITMAX,10)
    WRITE(IIM,340)
    WRITE(IIM,330) (QQ(J,IT),IT=1,ITMAX,10)
8000 CONTINUE
C
250 FORMAT(' ','IMPEDENCE AMPLITUDE NO. OF HARMONICS',I3)
260 FORMAT(' ',10E12.4)

```

```

270 FORMAT(' ','COSINE FOURIER PRESSURE COEFFICIENTS')
9270 FORMAT(' ','COSINE FOURIER FLOW COEFFICIENTS')
280 FORMAT(10F11.3)
290 FORMAT(' ','SINE FOURIER PRESSURE COEFFICIENTS')
9290 FORMAT(' ','SINE FOURIER FLOW COEFFICIENTS')
300 FORMAT(' ','PHASE ANGLE OF FOURIER PRESSURE COMPONENTS')
9300 FORMAT(' ','PHASE ANGLE OF FOURIER FLOW COMPONENTS')
310 FORMAT(10F8.3)
320 FORMAT(' ','PRESS VS TIME')
330 FORMAT(' ',13F10.3)
340 FORMAT(' ','FLOW VS TIME')
370 FORMAT(' ','AVERAGE PRESSURES,ALL TUBES,NODE 2')
380 FORMAT(' ',10F10.4)
390 FORMAT(' ','AVERAGE FLOWS,ALL TUBES,NODE 2')

```

C

```

IF (NPROFL.NE.0) THEN
  II=0
  ETA=0.0
  DO 9000 M=1,NVP
  DO 9100 I=1,30
  II=II+1
  RPLOT(II)=ETA
9100 ETA=ETA+.035
  ETA=0.0
9000 CONTINUE
  II=0
  DO 9500 IP=1,NPROFL
  DO 9510 M=1,NVP
  DO 9511 I=1,30
  II=II+1
9511 VCPLOT(II)=VCLPLT(IP,I,M)
9510 CONTINUE
  II=0
9500 CONTINUE
ENDIF
ENDIF

```

C

```

400 FORMAT(' TUB.DIA',10F7.3)
99400 FORMAT(' DWNSTRD',10F7.3)
410 FORMAT(' ','TER.LEN',10F7.1)
411 FORMAT(' ','TER.CAP',10F7.1)
420 FORMAT(' ','TER.DIA',10F7.3)
440 FORMAT(' ','TUBE NOS. WHERE VELOCITY PROFILE IS PLOTTED',10I5)
460 FORMAT(' ','NPSAVE',I3,2X,'TUBE NO. SPEC CALCS IIM FILE 80',15I4)
C 470 FORMAT(2X,'NTSTEN-THESE VESSELS HAVE LINEAR CALCS AT ALL T'S',5I5)
470 FORMAT(2X,'NTSTEN-THESE VESSELS HAVE NON-LINEAR P-A CALCS',5I5)
9470 FORMAT(2X,'NODSTEN',5I5)
99470 FORMAT(2X,'NTANUR-THESE VESSELS PROB HAVE LINEAR CALCS',5I5)
480 FORMAT(2X,'PSTEN FACTORS',5F5.2)
481 FORMAT(2X,'PANUR FACTORS',5F5.2)
482 FORMAT(2X,'PALFA FACTORS',5F5.2)
490 FORMAT(' ','STEADY FLOWS',10F10.4)
500 FORMAT(' ','QS(I)=' ,10F8.4)
510 FORMAT(2X,'CVTER(I) FOLLOWS',8E12.3)
511 FORMAT(2X,'CCV(I) FOLLOWS',8E12.3)
520 FORMAT(' ','CVTOT=' ,F12.7,5X,'CVTERM=' ,E10.3,5X,'RTOTSUM=' ,
1F12.4,5X,'SUM=' ,F12.6)

```

```

530 FORMAT(2X,'RTOT(I) FOLLOWS',10E10.4)
540 FORMAT(' ','NO. OF PRESS PTS IN TUBE',30I3)
550 FORMAT(' ','NO. OF FLOW PTS IN TUBE',30I3)
560 FORMAT(' ','IF IFRIC=0, WALL SHEAR IS NOT COMPUTED IFRIC=',I3)
565 FORMAT(2X,'FLAG(J)',50I2)
570 FORMAT(1X,'DT1,2=',2F8.6,1X,'XRTOT=',F5.2,1X,'PV=',F8.0,
*      'PO=',F8.0,1X,'RHO=',F5.2,1X,'MU=',F5.2,1X,
*      'PM=',F8.0,1X,'DELP=',F8.0,1X,'PRESI',F8.0)
580 FORMAT(' ','NTDIV,2,3,4=',4I5,2X,'NFORCE=',I3,2X,'NPTSFF=',I3,2X,
1'MULT FACTOR, INPUT SIGNALS',4F7.2)
590 FORMAT(' ','PHASE LAG FOR INPUT SIGNALS',12F7.1)
600 FORMAT(' ','HEART RATE=',F6.2,2X,'DX1,2=',2F6.3,2X,'TMAX=',F7.3,
*      2X,'NPERM=',I3,2X,'NPERL=',I3,2X,'NPERP=',I3,2X,'NPRIPP='
*      ,I5,2X,'TIMP=',I5)
99099 CONTINUE
      CALL TIME(ABCDE)
      WRITE(3,13132)NPER,ABCDE
      STOP
      END
      SUBROUTINE SUBSCR(NTUBES,FLAG,LLINK,NTJ,ISIGN,LQJ,LPTB,
*LPJ,KJ,LQ,LP,RLINK,MLINK,TERMS,NTNTOT,PLINK,ISOURCE,KSOURC)
      INTEGER TERMS,RLINK,LLINK,FLAG,NTJ,LQ,LP,MLINK,KJ,LPTB,LPJ,LQJ
      INTEGER ISIGN,PLINK,NTUBES,ISOURCE,KSOURC
      DIMENSION FLAG(NTUBES),LLINK(NTUBES),NTJ(NTUBES),LQ(NTUBES)
      DIMENSION LP(NTUBES),RLINK(NTUBES),PLINK(NTUBES),MLINK(NTUBES)
      DIMENSION ISIGN(NTUBES,4),LQJ(NTUBES,4),LPJ(NTUBES,4)
      DIMENSION LPTB(NTUBES,4),KJ(NTUBES,4),KSOURC(NTUBES)
      NTN=0
      DO 7777 J=1,NTUBES
        IF (FLAG(J).EQ.1) THEN
          NTN=NTN+1
          NTNTOT=NTN
C COUNT AND NUMBER EACH JUNCT, NTNTOT IS TOTAL NO.JUNC.
          IF(LLINK(J).GT.0) THEN
C CHECKS ONLY UNI- AND BI- FURCATIONS HERE, TRI-'S AT ELSE BELOW
C MOTHER VESSEL
            NTJ(NTN)=3
            ISIGN(NTN,1)=1.0
            LQJ(NTN,1)=LQ(J)
            LPTB(NTN,1)=LQ(J)
C IF ISOURCE INLET TUBE, ADJUST LENGTH
C IF (ISOURCE.EQ.0.AND.PLINK(J).NE.0) THEN
C LPTB(NTN,1)=LP(J)-1
C KSOURC(NTN)=7
C ENDIF
            LPJ(NTN,1)=LP(J)
            KJ(NTN,1)=J
            LLI=ABS(LLINK(J))
            KJ(NTN,2)=LLI
            IF (RLINK(LLI).NE.J) THEN
C DAUGHTER VESSEL, BEGINNING OF TUBE
              ISIGN(NTN,2)=-1.0
              LQJ(NTN,2)=1
              LPTB(NTN,2)=2
              LPJ(NTN,2)=1
              ELSE
C DAUGHTER VESSEL, END OF TUBE

```

```

    ISIGN(NTN, 2) = 1.0
    LQJ(NTN, 2) = LQ(LLI)
    LPTB(NTN, 2) = LQ(LLI)
    LPJ(NTN, 2) = LP(LLI)
    ENDIF
    IF (RLINK(J) .NE. 0) THEN
        KJ(NTN, 3) = RLINK(J)
        LLLJ = ABS(LLINK(RLINK(J)))
        IF (LLLJ .NE. J) THEN
            ISIGN(NTN, 3) = -1.0
            LQJ(NTN, 3) = 1
            LPTB(NTN, 3) = 2
CANVAS
        ELSE
CANVAS
            LQJ(NTN, 3) = LQ(RLINK(J))
            LPTB(NTN, 3) = LQ(RLINK(J))
            LPJ(NTN, 3) = LP(RLINK(J))
            ENDIF
            ELSE
            NTJ(NTN) = 2
            ENDIF
C CHECKS FOR TRIFURCATIONS HERE
        ELSE
            NTJ(NTN) = 4
            ISIGN(NTN, 1) = 1.0
            LQJ(NTN, 1) = LQ(J)
            LPTB(NTN, 1) = LQ(J)
C IF (ISOURCE.EQ.0.AND.PLINK(J).NE.0) THEN
C     LPTB(NTN, 1) = LP(J) - 1
C     KSOURC(NTN) = 7
C     ENDIF
            LPJ(NTN, 1) = LP(J)
            KJ(NTN, 1) = J
CANVAS
            KJ(NTN, 2) = LLK
CANVAS
            KJ(NTN, 4) = RLINK(J)
            IF (RLINK(LLK) .NE. J) THEN
                ISIGN(NTN, 2) = -1.0
                LQJ(NTN, 2) = 1
                LPTB(NTN, 2) = 2
                LPJ(NTN, 2) = 1
            ELSE
CANVAS
                LQJ(NTN, 2) = LQ(LLK)
                LPTB(NTN, 2) = LQ(LLK)
                LPJ(NTN, 2) = LP(LLK)
                ENDIF
                IF (MLINK(MLINK(J)) .NE. J) THEN
                    ISIGN(NTN, 3) = -1.0
                    LQJ(NTN, 3) = 1
                    LPTB(NTN, 3) = 2
                    LPJ(NTN, 3) = 1
                ELSE
                    ISIGN(NTN, 3) = 1.0
CANVAS

```

CANVAS

```
LPJ (NTN, 3) = LP (MLINK (J))
ENDIF
LLLJ = ABS (LLINK (RLINK (J)))
IF (LLLJ.NE.J) THEN
  ISIGN (NTN, 4) = -1.0
  LQJ (NTN, 4) = 1
  LPTB (NTN, 4) = 2
  LPJ (NTN, 4) = 1
ELSE
  ISIGN (NTN, 4) = 1.0
  LQJ (NTN, 4) = LQ (RLINK (J))
  LPTB (NTN, 4) = LQ (RLINK (J))
  LPJ (NTN, 4) = LP (RLINK (J))
ENDIF
ENDIF
ENDIF
7777 CONTINUE
RETURN
END
```